National Library of Canada
Canadian Theses Service
Ottawa, Canada
K1A 0N4

Bibliothèque nationale du Canada
Services des thèses canadiennes

CANADIAN THESES

THÈSES CANADIENNES

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED

LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE

Canada

NL 339 (r. 86/01)

National Library of Canada | Bibliothèque nationale du Canada

Canadian Theses Division | Division des thèses canadiennes

Ottawa, Canada
K1A 0N4

0-315-24741-X

PERMISSION TO MICROFILM — AUTORISATION DE MICROFILMER

• Please print or type — Écrire en lettres moulées ou dactylographier

Full Name of Author — Nom complet de l'auteur

Warren Hugh Finlay

Date of Birth — Date de naissance

Sept. 21, 1961

Country of Birth — Lieu de naissance

Canada

Permanent Address — Résidence fixe

Box 8644, Station L
Edmonton, Alberta
Canada T6C 4J4

Title of Thesis — Titre de la thèse

Fluid Particle Simulation of Viscous Flows

University — Université

University of Alberta

Degree for which thesis was presented — Grade pour lequel cette thèse fut présentée

M.Sc.

Year this degree conferred — Année d'obtention de ce grade

1984

Name of Supervisor — Nom du directeur de thèse

Dr. C.E. Capjack

Permission is hereby granted to the NATIONAL LIBRARY OF CANADA to microfilm this thesis and to lend or sell copies of the film.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

L'autorisation est, par la présente, accordée à la BIBLIOTHÈQUE NATIONALE DU CANADA de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans l'autorisation écrite de l'auteur.

Date

August 3/84

Signature

W. Finlay

NL-91 (4/77)

THE UNIVERSITY OF ALBERTA

Fluid Particle Simulation of Viscous Flows

by

© Warren Hugh Finlay

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF Master of Science

Electrical Engineering

EDMONTON, ALBERTA
Fall, 1984

August 8,1984

I, David M. Antoniuk, hereby authorize Warren H. Finlay to use all or any part of the following figures for the purpose of his M.Sc.(1984) thesis: figures 3.2, 3.7, 3.8, 4.1, 5.4 from my PhD (1983) thesis and figure 4 from the paper "Computer Simulation of Gas Transport in a Magnetically Stabilized Glow Discharge", J. Appl. Phys., vol. 55, no. 3, pp. 708-713, Feb. 1984, written by myself, C.E. Capjack and H.J.J. Seguin.

Signed: David M. Antoniuk

I, C.E. Capjack, hereby authorize Warren H. Finlay to use all or any part of the figure 4, mentioned above, for the purpose of his M.Sc. (1984) thesis.

Signed: C.E. Capjack.

I, H.J.J. Seguin, hereby authorize Warren H. Finlay to use all or any part of the figure 4, mentioned above, for the purpose of his M.Sc. (1984) thesis.

Signed: H. Seguin

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR Warren Hugh Finlay

TITLE OF THESIS Fluid Particle Simulation of Viscous Flows

DEGREE FOR WHICH THESIS WAS PRESENTED Master of Science

YEAR THIS DEGREE GRANTED FALL 1984

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

(SIGNED) W. Finlay

PERMANENT ADDRESS:

...Box 8644, Station L.....

...Edmonton, Alberta. T6C 4J4

...Canada.....

DATED ...July 31..... 1984

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled Fluid Particle Simulation of Viscous Flows submitted by Warren Hugh Finlay in partial fulfilment of the requirements for the degree of Master of Science.

.....Robert Kobosyan.....

.....E.M. Yate.....

Supervisors

.....C.E. Capjack.....

.....John Rogerson.....

Date....July 31, 1984.....

Dedication

To my loving wife Susan.

Abstract

Recent advances in computational plasma physics have been made by using a fluid particle approach to simulate the time dependent, compressible MHD equations. Inviscid fluids in periodic geometries have been successfully modelled by other researchers [2],[7] with such a particle method. The development of a method for the computer modelling of viscous, compressible fluids in nonperiodic geometries using a fluid particle approach is described in the following pages.

The fluid equations are first written in a Lagrangian form, which is conservative in a Lagrangian sense. The fluid particle approach then represents the fluid by a large number of "particles", which are not true point particles but rather have a finite spatial extent. Each finite particle has a velocity and temperature, so that the fluid quantities of velocity and temperature are obtained by accumulating the particle quantities on a fixed computational grid. A small scale smoothing is applied to the fluid velocity and temperature to reduce noise levels caused by random particle motion. The finite particles are given a Gaussian shaped mass density, so that the fluid mass density is obtained by summing the mass densities of nearby particles using a nearest grid point (NGP) approach. The fluid equations are then finite differenced on the grid, and the Lagrangian derivatives appearing in the fluid equations are then used to advance the particle velocities and

temperatures. Particle positions are changed according to particle velocities. A new method of assigning the pressure gradient force to the particles is developed, which resulted in a very significant reduction in the undesirable random particle motion that occurred with the inviscid simulations done by other researchers [2],[7]. The flow of a turbulent viscous fluid through a two dimensional channel is described analytically and experimentally, and a fluid particle method for simulating the channel is then developed. Several special considerations, required at the open flow boundaries, are described. The code results are compared to the analytic and experimental results. It is found that the multistreaming terms (which prevent particles from having velocities and temperatures much different from the fluid velocity and temperature, as used in [2]) must be used carefully in the case of channel flow. The source of the random particle noise is also discussed.

A magnetically stabilized gas discharge, which was being experimentally tested at the U of A [4], and was modelled numerically by Antoniuk in [4],[16] using a conventional fluid approach, is also modelled and the results compared to Antoniuk's experimental and computational results.

Acknowledgements

I wish to thank Dr. Capjack for his support; without his guidance and constant supply of creative ideas this project would not have been possible. I also would like to thank Dr. J.N. Leboeuf from the University of Texas, Austin for allowing access to his inviscid fluid particle simulation of the Kelvin-Helmholtz instablity. His code served as a starting point for the simulations done in this work and saved us a considerable amount of time. And finally, may God bless my family and friends for their love and support.

Table of Contents

Chapter		Page
1.	Introduction	1
2.	Mathematical and Computational Model	10
	2.1 Mathematical model	10
	2.2 Computational model	13
	2.2.1 The particle equations	16
	2.2.2 Finite differencing of the equations	18
	2.2.2.1 Modified gradp differencing	21
	2.2.2.2 Force conservative differencing	24
	2.2.3 Variable grids	25
	2.2.4 General conservative form of the equations	26
	2.2.5 Boundary conditions	27
3.	Experimental and Analytical Channel Flow	29
	3.1 Experimental and analytical relations for a few variables	29
	3.2 Definition of layers in channel flow	38
	3.3 Eddy size analysis	40
	3.4 Analysis of distance to reach steady state velocity profile	43
4.	Numerical Simulation of Channel Flow	47
	4.1 Boundary and initial conditions	47
	4.2 Open boundary considerations for channel flow	50

4.2.1
Streaming instability 50

4.2.2
Buffer zones 51

4.3 Other considerations 54

4.3.1
Gaussian averaging of v and T 54

4.3.2
Empty cells 56

4.4 Constraints on variables 59

4.5 The artificial drag term and noise caused by a NGP approximation 64

4.6 Results 73

5. Simulation of a Magnetically Stabilized Gas Discharge 82

5.1 Analytic considerations 82

5.2 Computational considerations 88

5.2.1
Boundary and initial conditions 88

5.2.2
Finite differencing considerations 93

5.3 Results and discussion 93

6. Concluding Remarks 104

6.1 Summary 104

6.2 Some further remarks 105

6.3 Future applications and considerations 106

References 109

List of Figures

Figure — Page

1.1 Square and guassian particle mass density shapes are two density profiles often assigned to particles.....7

2.1 Vp finite differencing. $\partial p/\partial x$ is evaluated at the crosses(x) and $\partial p/\partial x$ inside cell i,j is then a lineat interpolation in x between these two values as in equation (2.22). $\partial p/\partial y$ is evaluated at the circles(0) and is similarly interpolated.....23

3.1 A typical profile of along stream velocity U as a function of cross stream distance y for turbulent flow through a channel of cross stream width D.....31

3.2 Experimental plot of universal velocity profile $(U_c-U)/u_*$ versus normalized distance from the wall; obtained from [12, p. 482].....33

3.3 In a turbulent shear flow along a wall $\langle uv \rangle$ is less than zero, See page 36 for an explanation.....37

4.1 Ghost point configuration for channel simulation....49

4.2 Cells with no particles in them introduce a nonconservative effect into the normally conservative flux conservative finite differencing. A plus/minus sign above a cell signifies that the value of A from that cell is added/subtracted to/from a sum of the differencing over the grid. (1) No empty cells, (2) empty cells - no corrective action in the differencing and (3) Neumann treatment of empty cells.See p. 59 for a further explanation.....60

4.3 A velocity profile that results for channel flow if too large a value of the artificial viscosity coefficient a_m is used. The code results are indicated by the triangles, while the experimental results of Nikuradse and Donch [12, p. 482] are indicated by X and 0 respectively..68

4.4 Comparison of code velocity profile (triangles) with experimental data of Nikuradse (x) and Donch (0) from [12, p. 428] after 4200 time steps (4.47 ms).....74

4.5 Comparison of pressure profile (triangles) to analytic results (straight line).....76

4.6 A typical instantaneous flow pattern. This snapshot is for 3100 time steps (3.3 ms). The arrows are scaled to the maximum velocity, which is 186.6 m/s (Note: $\langle U \rangle_s = 112.2$ m/s).....77

4.7 Comparison of code shear stress profile (triangles) to known profile (straight line).....80

5.1 Magnetically stabilized transverse discharge geometry. This figure is taken from [4] with the author's permission.....83

5.2 Magnetic flux density of electromagnet - figure taken from [4] with author's permission.....84

5.3 Magnetic field intensity contours of electromagnet - figure taken from [4] with author's permission.....85

5.4 Gaussian shaped current density distribution - figure taken from [4] with author's permission.....87

5.5 Arrangement of ghost point boundaries on the finite difference mesh - figure taken from [4] with author's permission.....89

5.6 Contour plot of v_θ in m/s from particle code at t=4 ms (2000 time steps).....94

5.7 A Rayleigh-Taylor instability, as shown, would normally result in the gas discharge geometry where a less dense fluid contains a more dense fluid. See p. 95 for an explanation.....96

5.8 Contour plot of v_θ at t=4 ms obtained by Antoniuk [16] using ADI fluid code.....98

5.9 Typical particle code velocity flow plot. The arrows length are scaled with the magnitude of velocity. The maximum veloctiy is 36.4 m/s.....100

5.10 Contour plot of gas temperature T in degrees Kelvin from particle code at t=4 ms.....102

CHAPTER I

Introduction

Much of the nature of the physical world can be described by partial differential equations. Before the advent of the digital computer only a limited number of these equations had been solved to the extent of yielding useful physical solutions. However, in the past twenty years many solutions of these equations have been obtained by computer simulation.

In modelling any set of partial differential equations on a computer it is necessary to break the continuous nature of the partial differential equations into some discrete form. One way of doing this is the finite difference method. In the case where the differential equations involve the spatial and temporal variation of several quantities, then finite differencing begins by considering these quantities at only a finite number of points in time and space. These points form a temporal, spatial mesh or grid. The derivatives appearing in the partial differential equations are then approximated on the grid by the difference between the quantities at neighboring grid points. Using these approximated derivatives in the equations, the values of the variables involved in the equations can be evaluated at each grid point.

In the area of fluid dynamics there have been two basic approaches to applying the finite difference method to the governing partial differential equations of fluid motion.

The Eulerian codes (code is a term meaning computer program) use a fixed spatial finite difference grid, across which the fluid streams by. The grid is called Eulerian because it is stationary, so that the derivatives evaluated directly on the grid are partial or Eulerian derivatives. The other scheme involves moving the spatial grid with the fluid, resulting in a Lagrangian code. The term Lagrangian refers to the fact that the spatial grid distorts with the fluid, and the derivatives on the grid are Lagrangian derivatives (also known as convective, substantive or total derivatives).

One of the most severe drawbacks of the Eulerian method is the difficulty found in dealing correctly with the advective terms ($\Sigma_i\{v_i \partial A/\partial x_i\}$) in the total derivatives dA/dt appearing in the fluid equations. Instabilities producing, for example, negative densities can arise. These can be dealt with by such elaborate methods as flux corrected transport (FCT) [15], but this adds complexity as well as expense. As an alternative, the Lagrangian approach eliminates the advective derivative problems; however, in two and three dimensions, the spatial finite difference grid becomes so entangled that the method breaks down. As a result, in more than one dimension the Eulerian method is generally used.

In plasma physics, the laws governing a plasma can be written in a form similar to the normal fluid dynamics equations. For a plasma which is assumed to be

quasi-neutral (i.e. charge density $\rho \cong 0$), the magnetohydrodynamic (MHD) equations result. These are essentially an extension of the normal fluid equations to a conducting fluid, with the addition of Maxwell's electromagnetic equations. Alternatively, the plasma may be treated as a multi-species fluid; in which case, the electrons, ions, and neutrals are treated as separate, interacting fluids, each obeying equations with a similar form to the MHD equations. The Eulerian or Lagrangian finite difference method can be applied to the sets of equations resulting from the MHD or multi-species treatments; however, in addition to the finite difference difficulties mentioned above, these methods yield solutions to a plasma which has been assumed to obey a fluid picture. Many real plasmas, and most nonlinear plasma phenomena, do not obey the fluid type plasma equations and so the Eulerian or Lagrangian fluid finite difference schemes cannot be used with accuracy.

To overcome this difficulty of plasma simulations, one might consider dealing with individual electrons and ions, since their equations of motion are quite simple and one could solve self-consistently for the electric and magnetic fields based on the particle positions and velocities. Unfortunately, the number of particles involved in any practical system is far beyond the computational and memory capabilities of even the most advanced supercomputers. In the 1960's though, a simulation technique involving what has

been called "finite particles" arose [5],[6]. In most plasma simulations, these finite particles obey the same equations as a single real particle. In addition to having a position and velocity these macroparticles may also carry a charge, but since they are finite, this charge is spread out over a finite volume in space. Since each macroparticle represents a large number of real particles, the number of macroparticles required to model a realistic geometry is within the limits of today's computers.

Earlier finite particle plasma simulations advanced the finite particles explicitly in time. An explicit time difference scheme evaluates the values of the variables at time $(n+1)\Delta t$ based solely on their values at previous time steps. Explicit methods are conditionally stable: they must satisfy the condition that the time step Δt must be smaller than the time it takes for the fastest phenomena to cross one grid spacing. This requirement is called the Courant or Courant-Friedrichs-Lewy (CFL) stability criterion. For plasma simulations this is an extremely severe limitation since the fastest time scale is determined by either the ion plasma, electron cyclotron, or photon frequency, all of which usually yield time scales which are many orders of magnitude below the time scale of interest. This places the explicit simulation of most plasmas beyond the computation times available.

In recent years, the Courant limit has been removed by the development of implicit time advancement methods,

implicit meaning that the method of advancing the quantities at time $n\Delta t$ is dependent on the values one step ahead in time at $(n+1)\Delta t$. Clearly this requires some sophisticated computation, and indeed implicit particle simulations are presently largely limited to one and two dimensions because of this. Some recent hybrid codes have combined a finite particle method (to deal with the nonfluid electrons and ions) with a multifluid FCT Eulerian scheme (to take care of the cold background electrons and ions) [10].

The particle plasma methods in general are neither cheap nor simple, and for plasmas which can be assumed to obey the fluid type equations, or which are three dimensional, it is much easier and cheaper to use an Eulerian finite difference scheme. However, to remove the advective derivative difficulty associated with these codes (mentioned earlier), it is possible to employ a modified finite particle approach to the fluid equations, which is cost competitive with the Eulerian codes. The application of such a fluid particle method to viscous fluids was the purpose of this work. The fluid particle method applies a particle approach to the fluid equations, rather than to the particle equations which the particle plasma methods use.

When applying a finite particle method to the fluid equations, it is necessary to write the equations in their Lagrangian form. In this way the equations are written as $dA/dt=\ldots$ so that the quantity A can be carried by a finite particle. The change $(dA/dt)\Delta t$ is then very naturally

associated with the change in A as the finite particle moves from position $\underline{x}$ to position $\underline{x}=\underline{x}+\underline{v}\Delta t$. For the quasi-neutral or neutral fluid equations, rather than giving a macroparticle a spatial charge density it is more convenient for them to have a spatial mass density. The fluid is thus represented by a large number of macroparticles which may be thought of as having a cloud of mass or a mass density profile with origin at the macroparticle's center. In addition to having a position, velocity, and mass density, the particles may carry a temperature or any other quantity which has as its governing equation: $dA/dt=\ldots$. This approach to the fluid equations removes the serious advective difficulties found in traditional Eulerian finite difference codes by having the macroparticles carry the Lagrangian derivatives. In order to maintain a spatial resolution of approximately one grid spacing, the macroparticles are given a size of order the same size as a grid cell. Two common finite particle shapes are shown in figure 1.1. From the locations of the macroparticles the density of the fluid can be evaluated, and all fluid quantities can be evaluated at the Eulerian grid points as just an average of the temperatures and velocities of the macroparticles inside a cell. Electric and magnetic field quantities and current densities are not carried by the particles, but can be calculated on the grid by conventional finite difference methods [2],[7].

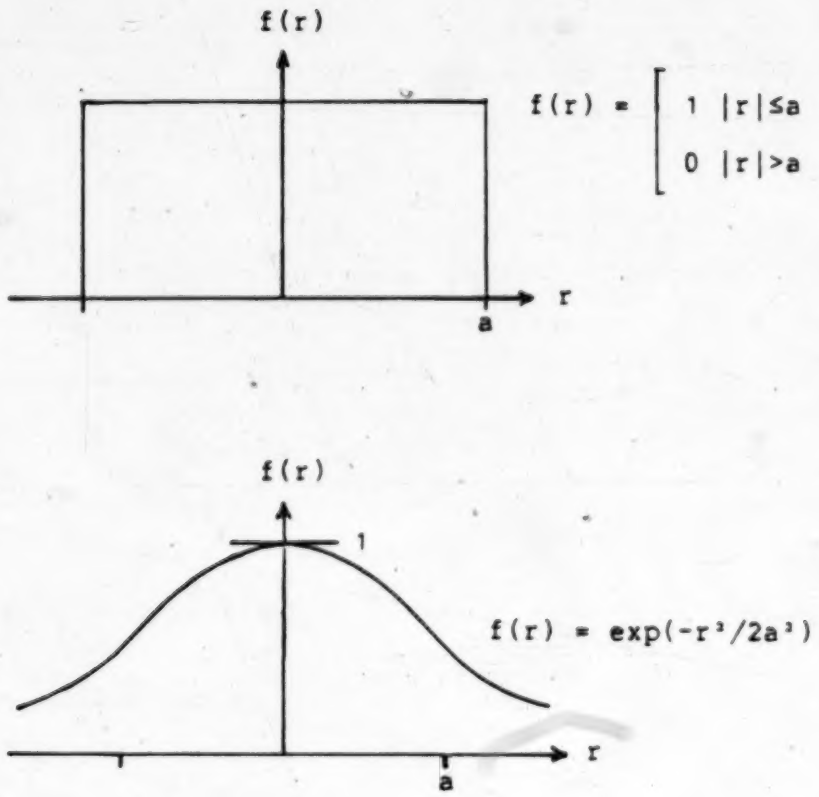

Figure 1.1 Square and guassian particle mass density shapes are two density profiles often assigned to particles

Previous finite particle approaches to the fluid equations have considered nonviscous (ideal) fluids in periodic geometries [2],[7]. (The term particle will from now on be used to mean macroparticle [finite particle] as is conventional in the literature.) Lebouef, Tajima, and Dawson in [2] use a nearest grid point (NGP) method to assign the particles to a cell. With the nearest grid point method a particle's position is approximated as being at the middle of the cell that the particle's center is in when evaluating fluid quantities. In addition, all particles whose centers are in this cell are pushed with the same force, which is calculated using conventional finite differencing of the fluid quantities. The NGP approach is computationally quite simple, but it leads to excessive random particle motion (noise) in the results. As an alternative, area weighted or particle in cell (PIC) methods [5],[7], which use square or rectangular particles, assign particle quantities to grids by considering the particle's area overlap in each cell. Forces are assigned to particles in a similar way. The PIC schemes reduce noise levels, and with second order PIC methods [8] noise levels and the numerical diffusion of quantities can be reduced further. However, considerable expense and complexity are added by employing area weighting methods. Subtracted dipole methods (SUD) [7] are also considerably more complex than the NGP method.

The intent of this project has been to develop a NGP particle method for simulating compressible, viscous fluids using the full instantaneous fluid (MHD) equations. A modified NGP approach was developed which yielded levels of random particle motion which were much lower than in previous NGP methods. In adding viscosity to the fluid equations, the assumption of a spatially periodic system was no longer possible and special considerations had to be given to open flow boundaries. Naturally the fast Fourier transform techniques used in evaluating the ∇p force for a periodic system [7] cannot be used with accuracy because of leakage and aliasing errors introduced by the nonperiodicity involved [22, p. 278]. Instead, pressure was obtained from the ideal gas law in which the fluid density was evaluted as an explicit sum of the macroparticle density contributions. A modified NGP approach was then used to assign the ∇p force to the particles. At open flow boundaries, special techniques were necessary to maintain a steady flow rate.

By removing the periodic boundary conditions and using the fully viscous fluid equations, many real problems can be dealt with using a particle approach. In addition, considerable simplicity is gained over other methods by using particles to deal with the total derivatives.

CHAPTER II

Mathematical and Computational Model

2.1 Mathematical model

In order to use a particle approach to model the fluid equations, the partial differential equations which govern a Newtonian fluid must be written in a Lagrangian form. In first defining the form of these equations it is most convenient to consider a neutral fluid. The equations for a MHD fluid (see Chapter 5) or a multifluid plasma are very similar and are easily formed from the neutral fluid equations.

Defining the velocity of the fluid as $\underline{v}(\underline{r},t)$ and its density by $\rho(\underline{r},t)$, then the momentum equation can be written as [1, p. 51]

$$\rho d\underline{v}/dt = \underline{B} + \nabla\cdot\check{\underline{q}} \qquad (2.1)$$

where $\underline{B}$ contains all body forces acting on the fluid, $\check{\underline{q}}$ is the stress tensor:

$$\check{\underline{q}} = [-p+(\beta-2\mu/3)\nabla\cdot\underline{v}]\underline{\underline{I}} + 2\mu\underline{\underline{S}}$$

and the total derivative is $d\underline{v}/dt = \partial\underline{v}/\partial t + \underline{v}\cdot\nabla\underline{v}$. Here, p is the local thermodynamic pressure; $\underline{\underline{I}}$ is the unit tensor; β is the second coefficient (or bulk) viscosity; μ is the first coefficient (or shear) viscosity; and $\underline{\underline{S}}$ is the strain tensor: $\underline{\underline{S}} = (\nabla\underline{v} + (\nabla\underline{v})')/2$, where t indicates the transpose, and $\nabla \equiv \Sigma_i\, \hat{e}_i\partial/\partial x_i$, $\hat{e}_i$ being a contravariant (or dual) base vector in the i'th direction. (Note that the gradient of a vector is a tensor.) For an ideal gas $\beta\equiv 0$ and for most

fluids $\beta \approx 0$ [1, p. 62], so that equation (2.1), with $\underline{B}=0$ and μ=constant, can be simplified to the well known form:

$$\rho d\underline{v}/dt = -\nabla p + \mu\nabla^2\underline{v} + \mu\nabla(\nabla\cdot\underline{v})/3 \qquad (2.2)$$

To ensure a total energy balance throughout the fluid, it is necessary to satisfy the First Law of Thermodynamics. This law states that for an elementary volume (ΔV) as it flows along its path:

$$dE_t/dt = Q' - W'$$

where dE_t is the increase in total energy of the volume in a time dt, Q' is the rate of heat transfer to the volume, and W' is the rate of work done by the volume [1, p. 265]. Here, Q' and W' are not written as dQ/dt and dW/dt, because this would indicate that Q and W could be transported by the fluid, which is not the case. E_t includes kinetic energy ($\rho\underline{v}\cdot\underline{v}/2$) and internal energy (e). Written for an elementary volume: $E_t = \rho\Delta V(e + \underline{v}\cdot\underline{v}/2)$. Potential energy terms are not included since they also appear in W' and so will cancel out [9, p. 72-74]. Letting $E_t=\rho e_t\Delta V$, with e_t being the total energy per unit mass, then

$$\rho\Delta V de_t/dt = \rho\Delta V(de/dt + d(\underline{v}\cdot\underline{v})/dt/2) \qquad [1, p. 266]$$

Using Fourier's law for heat flux, the rate of heat flow into the elementary volume is [9, p. 73]:

$$Q' = [\nabla\cdot(\kappa\nabla T)]\Delta V \qquad (2.3)$$

where κ is the constant of heat conductivity, and T is the temperature of the fluid. The rate of work done by the fluid is [9, p. 74]:

$$W' = [-\nabla\cdot(\underline{v}\cdot\underline{\tau})]\Delta V$$

$$W' = [\nabla\cdot(p\underline{v}) - \nabla\cdot(\underline{v}\cdot\underline{\underline{\tau}})]\Delta V \qquad (2.4)$$

where $\underline{\underline{\sigma}} = -p\underline{\underline{I}} + \underline{\underline{\tau}}$ so that $\underline{\underline{\tau}}$ is the viscous shear stress tensor $\underline{\underline{\tau}} = (-2/3\mu\nabla\cdot\underline{v})\underline{\underline{I}} + 2\mu\underline{\underline{S}}$. The First Law thus becomes

$$\rho de_t/dt = [\kappa\nabla^2 T - \nabla\cdot(p\underline{v}) + \nabla\cdot(\underline{v}\cdot\underline{\underline{\tau}})] \qquad (2.5)$$

In the systems considered in this project the fluids were ideal gases, so that the ideal gas law $p=\rho RT$ was obeyed. Other fluids could be modelled by a particle approach providing their equation of state (i.e. a relationship between p,e, and T) is known. Some other form of the energy equation (2.5), involving enthalpy for example, could be used instead; however, some equation of state is still needed.

For an ideal gas $de = Cv(dT)$, where Cv is the specific heat at constant volume. Equation (2.5) can thus be written:

$$dT/dt = [\kappa\nabla^2 T - \nabla\cdot(p\underline{v}) + \nabla\cdot(\underline{v}\cdot\underline{\underline{\tau}})]/\rho Cv - (\underline{v}\cdot d\underline{v}/dt)/Cv \qquad (2.6)$$

The term $\nabla\cdot(\underline{v}\cdot\underline{\underline{\tau}})$ includes the effects of viscous work $\underline{v}\cdot(\nabla\cdot\underline{\underline{\tau}})$ and viscous dissipation $\phi = \underline{\underline{\tau}}:\nabla\underline{v}$. The standard form [1, p. 267]

$$dT/dt = (\kappa\nabla^2 T - p\nabla\cdot\underline{v} + \phi)/\rho Cv$$

was not used because the pressure work term $p\nabla\cdot\underline{v}$ cannot be finite differenced conservatively, and led to large scale instabilities in the flow. It was also found that, even though the temperature was expected to be constant, equation (2.6) was necessary for the channel flow simulation, otherwise large nonphysical oscillations developed in the

flow. The discharge simulation described in chapter 5 was stable without the temperature equation, though the temperature was not expected to be constant there. Note in equation (2.6) that the term $\underline{v}\cdot d\underline{v}/dt$ is particularly suited to the particle method since $d\underline{v}/dt$ is known from equation (2.2). Note also that a mass conservation equation is not used with the particle approach since particles are not created or destroyed in a closed system, so that mass is naturally conserved.

2.2 Computational model

The Lagrangian equations of fluid motion for an ideal gas (equations [2.2] and [2.6]) can be modelled by considering the fluid as composed of many macroparticles, each with a spatial density profile. It is not correct to view the macroparticles as actual physical sections of the fluid, since the particles can only be properly described by the equations which relate the particle quantities to the fluid quantities and fluid equations.

In the systems considered herein, each particle has associated with it a position, a velocity, a mass density, and a temperature. To obtain the fluid velocity, temperature, and density, an Eulerian grid is used on which the fluid quantities are to be defined. The fluid temperature and velocity at the middle of a cell (i.e. at a grid point) are very simply defined as just an arithmetic average of the temperatures and velocities of the particles inside that

cell. To obtain the fluid density is more difficult because of the assumed finite extent of the particle densities. In this work the particles were given a Gaussian-shaped density profile (see fig. 1.1). It would be possible to sum the individual density contributions of all the macroparticles to obtain the fluid density at a point, but this would be very time consuming. Instead, a nearest grid point method (NGP) approximation is applied. Defining $n_{i,j}$ as the number of particles in cell i,j whose center is at $\underline{r}_{i,j}$, then the NGP approximation considers all particles in cell i,j as centered at $\underline{r}_{i,j}$. The density at any other other grid point m,n, located at $\underline{r}_{m,n}$, is then defined to be:

$$\rho(\underline{r}_{m,n},t) = \rho_0[\Sigma_{i,j} f(\underline{r}_{m,n}-\underline{r}_{i,j})n_{i,j}(\underline{r}_{i,j},t)]/ [\Sigma_{i,j} f(\underline{r}_{m,n}-\underline{r}_{i,j})n_0] \qquad (2.7)$$

where ρ_0 is the initial fluid density associated with n_0 particles per cell; the sums $\Sigma_{i,j}$ are over all grid points; and $f(\underline{r} - \underline{r}_{i,j})$ describes the spatial extent of a macroparticle density profile. For R^n, or for an n-dimensional periodic system, the denominator of equation (2.7) is written as $2\pi^{n/2}a^n n_0$ [6]. For a finite system, however, it is necessary to use the normalization factor given in the denominator of equation (2.7) since this accounts for the fact that near boundaries the numerator is less in value. To obtain a smooth fluid picture from the particles, a Gaussian profile of width 'a' was used for $f(\underline{r} - \underline{r}_{i,j})$ i.e.

$$f(\underline{r}_{m,n}-\underline{r}_{i,j}) = \exp[-(x_{m,n}-x_{i,j})^2/(2a^2) - (y_{m,n}-y_{i,j})^2/(2a^2)] \qquad (2.8)$$

where 'a^2' was set equal to between one-half and one times Δ^2, where Δ is the grid spacing.

The fluid quantities are thus known at each grid point so that finite differencing can be used to evaluate the right hand side of equations (2.2) and (2.6), and approximate the Lagrangian derivatives $d\underline{v}/dt$ and dT/dt at each grid point i,j. As each particle k is pushed a distance $\Delta\underline{x}_k = \underline{v}_k\Delta t$, its velocity and temperature are then changed by $\Delta\underline{v}_k = (d\underline{v}/dt)_{i,j}\Delta t$, $\Delta T_k = (dT/dt)_{i,j}\Delta t$ where i,j indicates the NGP to the particle. With this new configuration of particles the fluid quantities can be recalculated and the particles pushed again, etcetera.

It is important to note that the particles do not physically extend in space, since there are two major difficulties in such an interpretation. First, different parts of an actual spatial particle would be pushed or heated by different amounts and rates, corresponding to the particle's mass in different cells so that distortion would occur very quickly just as in the Lagrangian finite difference codes. Secondly, there would be an inconsistency in assigning the fluid quantities $(d\underline{v}/dt)_{i,j}$ and $(dT/dt)_{i,j}$ to the particles in cell i,j. This is because these derivatives are based on the spatial derivatives of the fluid, so that assigning these to a particle implies that a truly finite particle's velocity and temperature would vary in space. This would add a considerable amount of

complexity to the problem. The particles are really then just a means of carrying the Lagrangian derivatives on an Eulerian grid.

2.2.1 The particle equations

The actual form of the equations for the change in the kth particle's position, velocity and temperature are:

$$d\underline{x}_k/dt = \underline{v}_k \tag{2.9}$$

$$d\underline{v}_k/dt = [\underline{L}(\underline{v},T,\rho)_{i,j}]/\rho_{i,j} \tag{2.10}$$

$$dT_k/dt = [E(\underline{v},T,\rho)_{i,j}]/\rho_{i,j}Cv - (\underline{v}_k \cdot d\underline{v}_k/dt)/Cv \tag{2.11}$$

$\underline{L}_{i,j}$ and $E_{i,j}$ indicate the values of the right hand side of equations (2.2) and (2.5) respectively, finite differenced at the cell i,j that the particle is in.

All quantities in $\underline{L}$ and E should be staggered by a half time step from the values used in the approximations to $d\underline{v}/dt$, dT/dt [2, p. 405]. A correct time-centered scheme for equations (2.9)-(2.11) written at $t=n\Delta t$ is:

$$\underline{x}_k^{n+1} - \underline{x}_k^n = \underline{v}_k^{n+1/2}\Delta t \tag{2.12}$$

$$\underline{v}_k^{n+1/2} - \underline{v}_k^{n-1/2} = [\underline{L}(\underline{v}^{(n)},T^{(n)},\rho^n)_{i,j}]\Delta t/\rho_{i,j}^n \tag{2.13}$$

$$T_k^{n+1/2} - T_k^{n-1/2} = [E(\underline{v}^{(n)},T^{(n)},\rho^n)_{i,j}]\Delta t/\rho_{i,j}^n Cv - (\underline{v}_k^{n+1/2}+\underline{v}_k^{n-1/2})\cdot(\underline{v}_k^{n+1/2}-\underline{v}_k^{n-1/2})/2Cv \tag{2.14}$$

where the fluid quantity $\underline{v}^{(n)}$ is

$$(\underline{v}^{(n)})_{i,j} = [1.5\Sigma_{k(i,j)}\underline{v}_k^{n-1/2} - 0.5\Sigma_{k(i,j)}\underline{v}_k^{n-3/2}]/(n_{i,j})^n \tag{2.15}$$

- similarly for $T^{(n)}$. Note that equation (2.15) is valid only for Δt being a constant. The sums $\Sigma_{k(i,j)}$ are over all particles in the cell i,j, and $(n_{i,j})^n = (\Sigma_{k(i,j)}1)^n$ is the

number of particles in the cell i,j at time t=nΔt. This peculiar form for $\underline{v}^{(n)}$ can be written as $(\underline{v}^{(n)})_{i,j} = (\Sigma_{k(i,j)}\underline{v}_k^{(n)})/(n_{i,j})^n$ which is simply a linear approximation to the average of the particle velocities in cell i,j at t=nΔt. The form of $(\underline{v}_k \cdot d\underline{v}_k/dt)^n$ in equation (2.14) uses the value of $(d\underline{v}_k/dt)^n$ from equation (2.13). An interpolation is used here for the kth particle velocity $\underline{v}_k^n$, rather than the extrapolation which had to be used for the fluid velocity $(\underline{v}^{(n)})_{i,j}$ in $\underline{L}$ and E. This is because, at the point in the code where equation (2.14) is used to heat or cool the particles, $\underline{v}_k^{n+1/2}$ is known from equation (2.13). In calculating $\underline{L}$ and E however, $\underline{v}_k^{n+1/2}$ was not known yet and so extrapolation for $\underline{v}^{(n)}$ (2.15) was required. In (2.13) and (2.14), ρ^n is the value of the density based on the positions of the particles $\underline{x}_k^n$ at t=nΔt. This time-centering scheme improves the accuracy of the values of d/dt by assuming the quantities vary linearly between time steps.

With the form of equations (2.12)-(2.14) it is possible for a particle velocity or temperature to be quite different from the fluid velocity or temperature of the cell the particle is in. This is referred to as multistreaming, and can be removed by adding an artificial drag term [2] to the particle equations (2.13) and (2.14):

$$\underline{v}_k^{n+1/2} - \underline{v}_k^{n-1/2} = [\underline{L}_{i,j}^n]\Delta t/\rho_{i,j}^n - a_m(\underline{v}_k^{n-1/2} - \langle\underline{v}\rangle^n) \quad (2.16)$$

$$T_k^{n+1/2} - T_k^{n-1/2} = [E_{i,j}^n]\Delta t/\rho_{i,j}^n Cv - (\underline{v}_k \cdot d\underline{v}_k/dt)^n/Cv - a_t(T_k^{n-1/2} - \langle T\rangle^n) \quad (2.17)$$

where

$$\langle \underline{v} \rangle^n = (\Sigma_{k(i,j)} \underline{v}_k^{n-1/2})/(n_{i,j})^n \qquad (2.18)$$

(similarly for $\langle T \rangle^n$) and a_m and a_t are constants (referred to collectively as "a") $0 \le a \le 1$. The value of a_m and a_t used must be examined carefully, as discussed in section 4.5. $\langle \underline{v} \rangle^n$ and $\langle T \rangle^n$ appear similar to $\underline{v}^{(n)}$ and $T^{(n)}$ from (2.15), but the peculiar mix of time steps in (2.18) is necessary so that, when summed over all particles in a cell, the drag terms vanish. That is, when the last term on the right hand side of equation (2.16) is summed over all particles in cell i,j, the result is $-a_m(\langle \underline{v} \rangle^n - \langle \underline{v} \rangle^n) = 0$. This ensures that in any cell no net force appears from the artificial drag terms [2].

2.2.2 Finite differencing of the equations

Consider now the way in which to approximate the terms appearing on the right hand side of equations (2.2) and (2.5), which comprise $\underline{L}_{i,j}$ and $E_{i,j}$ in the particle equations (2.16) and (2.17). The fluid is divided into cells by the Eulerian grid; at the middle of each cell i,j, the fluid quantities $\underline{v}_{i,j}^{(n)}$ and $T_{i,j}^{(n)}$ are known, and these can be considered as the average value of the fluid quantities for that cell. The method chosen to approximate the derivatives should be conservative between time steps in the total momentum and energy of the system. That is, $\Sigma_k m_k d\underline{v}_k/dt$ (where the sum is over all particles) must be nonzero only due to momentum being added or lost at the boundaries, assuming there are no body forces. If there are

body forces, then this statement must still be true if the contributions of the body force $\underline{B}$ to the sum are left out. Similarly in energy, $\Sigma_k m_k (de_i/dt)_k$ must be nonzero only due to energy transfer at the boundaries, if the contributions from any internal body heat sources (for example Joule heating in an MHD fluid) are left out of the sum. (Note: strictly speaking it is $\Sigma_{i,j}[\rho \partial \underline{v}/\partial t]_{i,j}$, and $\Sigma_{i,j}[\rho \partial e_i/\partial t]_{i,j}$; where the sums are over all grid points and the derivatives are partial, which should be conserved; but this is not possible with a macroparticle scheme. By using total derivatives in the sums, the system will be conservative in a Lagrangian rather than an Eulerian sense.) Using equation (2.16) then:

$$\Sigma_k m_k d\underline{v}_k/dt = \Sigma_{i,j}\Sigma_{k(i,j)}\underline{L}(\underline{v},T,\rho)_{i,j} m_k/\rho_{i,j} \qquad (2.19)$$

The sum $\Sigma_{k(i,j)}$ is over all particles k in the cell i,j. The dummy index k appearing on the right hand side of (2.19) has no connection with the dummy index appearing on the left hand side. The mass of each particle m_k on a uniform grid is $m_k = \rho_{i,j}\Delta^3/n_{i,j}$, where Δ is the grid spacing. So (2.19) becomes:

$$\Sigma_k m_k d\underline{v}_k/dt = \Delta^3 \Sigma_{i,j}\underline{L}(\underline{v},T,\rho)_{i,j} \qquad (2.20)$$

The energy equation can be treated in the same way to yield:

$$\Sigma_k m_k (de_i/dt)_k = \Delta^3 \Sigma_{i,j} E(\underline{v},T,\rho)_{i,j} \qquad (2.21)$$

From equations (2.20),(2.21) it is apparent that we need a method of finite differencing which, when applied to the right hand side of equations (2.2) and (2.5), yields sums $\Sigma_{i,j}\underline{L}$ and $\Sigma_{i,j}E$ that are dependent only on the values of the

fluid quantities at the boundaries. In cartesian coordinates such a method exists and will be termed "flux conservative" differencing [3]. There is no physical interpretation to the name. This method can be applied to $\underline{L}$ and E, but to apply it to the viscous terms in E is tedious. The term $\nabla\cdot(\underline{v}\cdot\underline{\underline{\tau}})$, involving viscous work and viscous dissipation in equation (2.5), was expected to be small in comparison to the other terms and was not of prime importance in the geometries considered; therefore, a nonconservative form was used for it. All other terms in E and $\underline{L}$ were differenced by flux conservative methods. In noncartesian coordinates it is not possible to difference all terms flux conservatively; however, a "force conservative" differencing can be introduced in an attempt to have a stable form of differencing.

The principle of flux conservative differencing is to have the values of $\underline{L}$ and E cancelled between cells in the Eulerian grid. For example, the term $\partial(FG)/\partial x_1$, where F and G represent fluid quantities or position, can be differenced on a uniform grid as:

$$[\partial(FG)/\partial x_1]_{i,j} \cong (F_{i+1/2,j}G_{i+1/2,j} - F_{i-1/2,j}G_{i-1/2,j})/\Delta$$

where $F_{i+1/2,j} = (F_{i+1,j}+F_{i,j})/2$, and i,j refers to the position of the grid point i,j on the mesh. This is easily seen to satisfy the form of flux conservative differencing when summed over the entire mesh. When G is a derivative: $G = \partial H/\partial x_1$, this becomes:

$$[\partial(F\partial H/\partial x_1)/\partial x_1]_{i,j} \cong [F_{i+1/2,j}(H_{i+1,j}-H_{i,j}) - F_{i-1/2,j}(H_{i,j}-H_{i-1,j})]/\Delta^2$$

Some other flux conservative finite difference approximations are [3, p. 59]:

$$(\partial F/\partial x_1)_{i,j} \cong (F_{i+1,j}-F_{i-1,j})/2\Delta$$

$$(\partial^2 F/\partial x_1^2)_{i,j} \cong (F_{i+1,j}-2F_{i,j}+F_{i-1,j})/\Delta^2$$

$$[\partial(\partial G/\partial x_1)/\partial x_2]_{i,j} \cong [(\partial G/\partial x_1)|_{i,j+1}-(\partial G/\partial x_1)|_{i,j-1}]/2\Delta$$

$$\cong (G_{i+1,j+1}-G_{i-1,j+1}-G_{i+1,j-1}+G_{i-1,j-1})/4\Delta^2$$

2.2.2.1 Modified gradp differencing

In differencing the ∇p term in equation (2.2), ordinary flux conservative differencing proved unsatisfactory because it led to large scale clumping of the particles, producing large variations in density over only a few grids. Particle oscillations were created by these density fluctuations and large scale random velocities resulted; this is the source of the very undesirable noise levels found in previous NGP codes. As a simple example of this, consider a cell which has a large number of particles and thus a high mass density. Suppose this cell is surrounded by cells which all have the same number of particles, and this number is less than that of the middle cell. Then the particles in the middle cell will erroneously see no ∇p force since: $(p_{i,j+1}-p_{i,j-1})/2\Delta = (p_{i+1,j}-p_{i-1,j})/2\Delta = 0$; this is clearly in error since in fact the particles in this cell should feel a ∇p force which pushes them away from the middle of the cell in order to fill in the lower density region surrounding them.

The difficulty introduced by a conventional finite differencing of ∇p evaluated directly at the cell center was removed by evaluating ∇p at the midpoints between neighboring cell centers, and then linearly interpolating between these values based on a particle's position in the cell. For example, in cartesian coordinates x and y, for the kth particle in cell i,j (see figure 2.1):

$$(\partial p/\partial x)_k = \partial p/\partial x|_{i-1/2,j} + (\partial p/\partial x|_{i-1/2,j} - \partial p/\partial x|_{i-1/2,j})(x_k - x_{i-1/2,j})/\Delta \quad (2.22)$$

where $\partial p/\partial x|_{i-1/2,j} = (p_{i,j} - p_{i-1,j})/\Delta$, and similarly for the other derivative term. $(x_k - x_{i-1/2,j})$ is the distance to the right of $x_{i-1/2,j}$, the midpoint between cell centers i-1,j and i,j. $\partial p/\partial y$ is evaluated in a similar manner between midpoints i,j+1/2 and i,j-1/2. All forces could be assigned to the particles in this way, however it was found sufficient to deal with just the ∇p term in this way.

Note that equation (2.20) can no longer be strictly satisfied, since the value of ∇p in $\underline{L}$ for each particle in cell i,j is no longer the same. Equation (2.20) must now be written:

$$\Sigma_k m_k d\underline{v}_k/dt = \Delta^3 \Sigma_{i,j} \Sigma_{k(i,j)} [(-\nabla p)_k/n_{i,j}] + \Delta^3 \Sigma_{i,j} (\underline{L} + \nabla p)_{i,j}$$

The second sum on the right hand side vanishes since terms other than ∇p in $\underline{L}$ are differenced flux conservatively. The first sum however is not identically zero unless ∇p is the same for all $n_{i,j}$ particles in the cell i,j. This should not cause any instabilities to form however since when averaged over many time steps the sum will be approximately

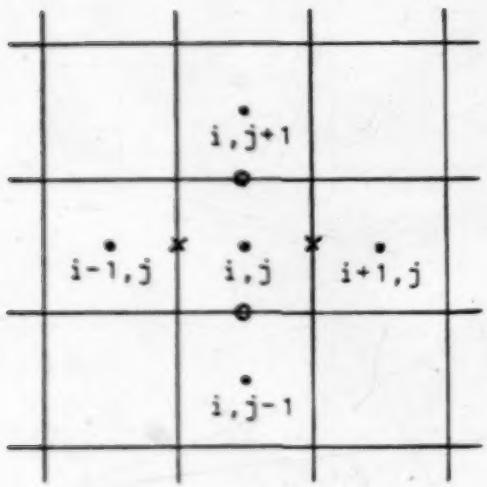

Figure 2.1 ∇p finite differencing. $\partial p/\partial x$ is evaluated at the crosses(x) and $\partial p/\partial x$ inside cell i,j is then a linear interpolation in x between these two values as in equation (2.22). $\partial p/\partial y$ is evaluated at the circles(0) and is similarly interpolated.

zero for steady state conditions. This is because the individual $(\nabla p)_k$ are obtained by flux conservative means. To show that the sum is approximately zero, consider for simplicity just the x direction in cartesian coordinates. Then the particle k from equation (2.22) contributes $(\partial p/\partial x)_k$ when it occupies position x_k. Assuming steady state conditions, then if at some other time a particle occupies the position $x_m=-x_k$, and at some other time another particle occupies the position $x_l=-x_k+2\Delta$, then the contribution to the ∇p sum from particle k will be zero. Because there are many particles per cell these events are likely to be approximately true after relatively few time steps.

2.2.2.2 Force conservative differencing

As mentioned above, some terms in $\underline{L}$ and E cannot be differenced flux conservatively when $\underline{L}$ and E are written in noncartesian coordinates. All such terms are of the form $F\partial G/\partial x$, and cannot be differenced so that their sum over the grid vanishes. The value of $F\partial G/\partial x$ is instead approximated at a cell center by an average of a finite difference approximation at the midpoints between cell centers, i.e. a difference of the form

$$F\partial G/\partial x = [F_{i+1/2,j}(G_{i+1,j}-G_{i,j})/\Delta + F_{i-1/2,j}(G_{i,j}-G_{i-1,j})/\Delta]/2$$

is used. This is done in order to have the values of $F\partial G/\partial x$ continuous at the interfaces of the cells. The term "force" has no physical significance in the expression "force differencing".

Note that a term of the form $\partial(F\partial G/\partial x_1)/\partial x_2$ is flux conservative, even though the term $F\partial G/\partial x_1$ is differenced as a force conservative term, i.e.

$$\partial(F\partial G/\partial x_1)/\partial x_2 \cong \{[F_{i,j+1/2}(G_{i,j+1}-G_{i,j})/\Delta + F_{i,j-1/2}(G_{i,j}-G_{i,j-1})/\Delta]_{i,j+1/2}$$
$$- [F_{i,j+1/2}(G_{i,j+1}-G_{i,j})/\Delta + F_{i,j-1/2}(G_{i,j}-G_{i,j-1})/\Delta]_{i,j-1/2}\}/2\Delta$$

where

$$F_{i+1/2,j+1/2} = [(F_{i+1,j+1}+F_{i,j+1})/2 + (F_{i+1,j}+F_{i,j})/2]/2$$

2.2.3 Variable grids

One can apply the above method of finite differencing to a variable grid [3, p. 60]; however, the macroparticle method becomes more expensive in this case, not only because of an increase in the complexity of the finite differencing and calculating the density on a variable grid, but also because there is a minimum number of particles per cell required to maintain a reasonable accuracy. Since the number of particles per cell is proportional to a cell area, then the larger cells require more particles per cell than the minimum. This adds to the computational work necessary since the larger cells have more particles than is necessary for good accuracy. Using a variable grid to obtain better resolution in certain areas may thus not be as cost effective as in conventional fluid codes. This is especially true if most cells are of a size significantly larger than the smallest cell. A uniform grid was used for all problems considered in this project since there was

little need for a variable grid for the geometries under consideration.

2.2.4 General conservative form of the equations

In order to be able to satisfy the condition that the change in total momentum and total energy depend only on fluid quantities at the boundaries, the form of the momentum equation to be finite differenced (2.2) must be of the general conservative form (ignoring body forces as mentioned earlier):

$$\rho d\underline{v}/dt = \mathrm{div}(\underline{\underline{M}}) \tag{2.23}$$

where $\mathrm{div}(\underline{\underline{M}}) \equiv \underline{\underline{M}}\cdot\nabla$ and the derivatives operate on $\underline{\underline{M}}$. Note that for a symmetric tensor $\mathrm{div}(\underline{\underline{M}}) = \nabla\cdot\underline{\underline{M}}$ and all tensors in the previous equations are symmetric. Integrated over the entire volume (V) this equation yields:

$$\int(\rho d\underline{v}/dt)dV = \int \mathrm{div}(\underline{\underline{M}})dV = \int \underline{\underline{M}}\cdot\hat{n}dS \tag{2.24}$$

where S is the surface enclosing the system and $\hat{n}$ is the unit normal to the surface. In this way, the only physical sources or sinks of $\rho d\underline{v}/dt$ result from the system boundaries. Similarly, the energy equation (2.5) must be of the form (ignoring internal heat sources as mentioned earlier):

$$\rho de_t/dt = \nabla\cdot\underline{A} \tag{2.25}$$

where $\underline{A}$ is a vector, so that

$$\int(\rho de_t/dt)dV = \int \underline{A}\cdot\hat{n}dS \tag{2.26}$$

Although in noncartesian coordinates the sums on the right hand side of (2.20) and (2.21) cannot be made dependent only

on the boundary conditions, it is still desirable to have the equations in conservative form ([2.23],[2.25]) so that the integral equivalent of (2.20),(2.21) ([2.24],[2.26]) results in there being no nonphysical sources or sinks of momentum or energy.

2.2.5 Boundary conditions

The boundary conditions applied to the finite differenced form of L and E are the "ghost point" or "guard point" conditions. That is, the boundaries of the fluid system are made to lie midway between two grid points, one interior and the other exterior to the system, the exterior grid points being where the boundary conditions are defined. The values of the fluid quantities at the fictitious exterior grid points are defined so that the correct boundary conditions for the problem being considered are satisfied. For example, if the fluid velocity is to be zero on a containing wall, then the ghost point fluid quantities would be the negative of those interior to the wall. To have a Dirichlet condition at a boundary, then the value of the quantity at an exterior grid point is set so that the average of the ghost point value and the value at the grid point interior to the boundary is the Dirichlet value. To have a zero derivative of some quantity at a boundary, then the ghost point value is made equal to the interior value. All boundary conditions for the fluid equations can be satisfied in this manner. In addition, at a rigid wall the

particles are elastically reflected.

CHAPTER III

Experimental and Analytical Channel Flow

In developing the viscous fluid particle method, a very well studied geometry was needed for simulation so that comparison of code results to experimental results could be done reliably. For this reason, the flow of a fluid between two very large smooth planar plates, known as smooth channel flow, was selected. The quantity of available experimental data and analytical results for channel flow is large, and only a small amount will be discussed here. In verifying the code results, several important parameters which characterize the flow were chosen for comparision, although many more less critical parameters could also be compared.

In describing the flow of any fluid it is important to distinguish between laminar flow, which is free from any turbulence, and turbulent flow. According to Hinze [11, p. 2], "Turbulent fluid motion is an irregular condition of flow in which the various quantities show a random variation with time and space coordinates, so that statistically distinct average values can be discerned." Most realistic flows are turbulent and in the following discussion only turbulent flows will be considered.

3.1 Experimental and analytical relations for a few variables

Since there is a random component to turbulence, a turbulent flow can only be described consistently in terms

of time averaged quantities. It was necesary to time average the code results to compare to other results since the instantaneous fluid equations were simulated. Note that by time averaging the fluid equations (2.2) and (2.6), the so called Reynolds averaged equations for the time average fluid quantities can be obtained [11, p. 21]. These equations contain too many variables to be solved without some assumptions being made, and are not suitable for a particle simulation because the particles behave in an instantaneous, not a time averaged manner. The time averaged quantities can always be obtained from the instantaneous values by a straight-forward time averaging. In most literature a time averaged quantity is indicated by a line above the instantaneous quantity. Unfortunately, due to word processing limitations this notation could not be used. Instead, in the following discussion, the triangular brackets < > enclosing a quantity will indicate the time averaged value of that quantity unless otherwise specified. The time average is assumed to be taken for a long enough period that the time averaged quantity is no longer a function of time. A capital letter U will be used to indicate $\langle v_x \rangle$, the time averaged along stream velocity component; and a capital letter P will be used to indicate $\langle p \rangle$, the time averaged thermodynamic pressure. Other conventions will be described as they are encountered.

For channel flow a typical time averaged along stream velocity profile is shown in figure 3.1. Note that the

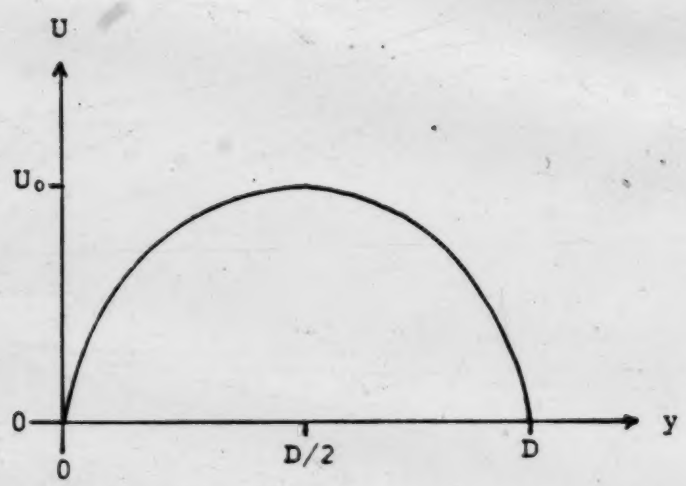

Figure 3.1 A typical profile of along stream velocity U as a function of cross stream distance y for turbulent flow through a channel of cross stream width D.

profile is symmetric about its centerline. Channels of different width, different midstream velocity, or different fluids will have different profiles; however, it is known that if the quantity $(U_o-U)/u*$ is plotted against cross stream distance $2y/D$ where D is the width of the channel, then a profile which is nearly universal for all turbulent channel flows is obtained [1, p. 587]. $u*$ is called the wall-friction velocity and is a characteristic constant of the flow defined by:

$$(u*)^2 \equiv \tau_o/\rho \qquad (3.1)$$

where τ_o is the value of the time averaged x-direction shearing stress felt at the wall; U_o is the time averaged midstream velocity; and U is the time averaged along stream velocity. This profile is somewhat different than that obtained for a flat plate, but is very similar to that for a round pipe. When very near the wall, the velocity profiles for the flat plate, round pipe, and channel are the same [11, p. 717]. Analytic functions for $(U_o-U)/u*$ can be obtained [1, p. 586],[12], but these are always verified by comparison to experimental results so that experimental results were used for comparison of the code results. The data of Nikuradse and Dönch obtained from [12, p. 482] was used and is plotted in figure 3.2.

It is also possible to define velocity profiles $U/u*$ which are functions of $yu*/\nu$, where $\nu=\mu/\rho$, but these are only valid for a fraction of the channel width. Another analytic function which is somewhat approximate, but is

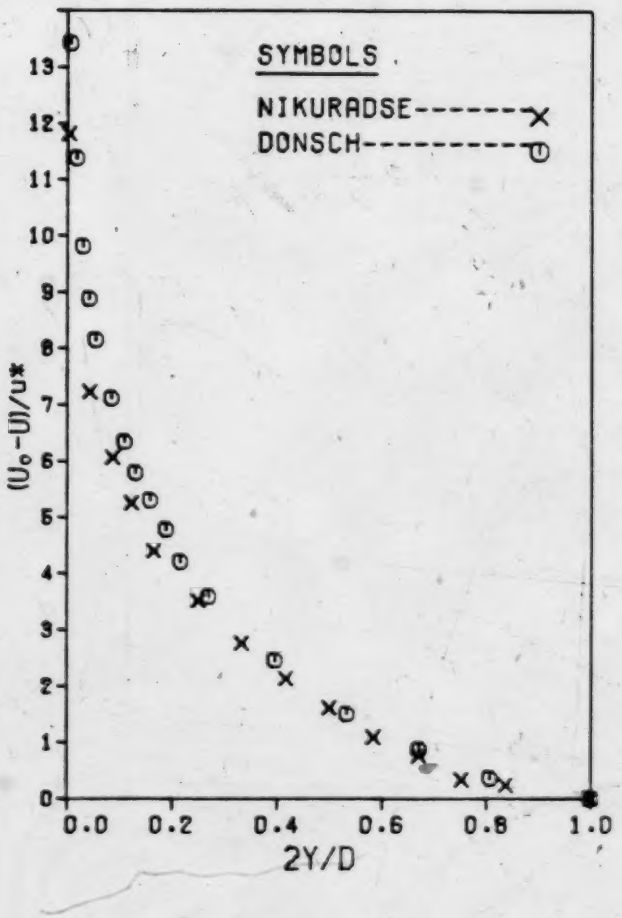

Figure 3.2 Experimental plot of universal velocity profile $(U_o-U)/u^*$ versus normalized distance from the wall; obtained from [12, p. 482]

often used because of its simplicity, is the power law [7, p. 723]

$$U/U_o = (2y/D)^{1/n} \qquad (3.2)$$

where n is a parameter which varies with the Reynolds number (based on the midstream velocity) $Re_o \equiv \rho D U_o/\mu$. The Reynolds number represents the ratio of inertial forces to viscous forces [1, p. 13], and flows which have the same Reynolds number are dynamically similar.

In order to compare the code results to the data of Dönch and Nikuradse (fig. 3.2), it is necessary to obtain the value of u* for the flow conditions being simulated. This can be done by using the definition of the resistance coefficient of pipe flow [1, p. 612]:

$$\Delta P/L = \lambda \rho (\langle U \rangle_s)^2/2h \qquad (3.3)$$

where λ is the resistance coefficient; ΔP is the drop in P, the time averaged pressure, over an along stream distance L; $\langle U \rangle_s$ is the value of U averaged across the channel; and h is the hydraulic diameter h=4A/C, A denoting the cross-sectional area and C denoting the wetted perimeter - for channel flow h=2D.

Now, the forces acting on a section of fluid of length L, width D, and of unit depth must be zero for equilibrium flow, so that the total shear force due to τ_o acting on the fluid at the wall must equal the total pressure force acting over the cross-section due to the pressure drop ΔP. Note that P, the time averaged pressure, cannot vary over the cross section, for obvious physical reasons. In the form of

an equation this statement of zero total force can be written:

$$D\Delta P = 2L\tau_0 \tag{3.4}$$

By combining (3.1), (3.3) and (3.4) then:

$$u^* = \langle U \rangle_A \lambda^{1/2}/\sqrt{8} \tag{3.5}$$

λ is a function of the Reynolds number $Re_d \equiv D\rho\langle U\rangle_A/\mu$ and can be obtained graphically from a "Moody diagram", which is essentially an experimental plot of λ versus Re_d; or an analytic approximation to the Moody diagram can be used. The latter route was chosen and λ was approximated as [13]:

$$\lambda = [0.868\ell n(Re_d/(1.964\ell nRe_d - 3.82))]^{-2} \tag{3.6}$$

for smooth channel flow. In this way, since $\langle U\rangle_A$ is easily obtained from the code results of U(y), the value of u^* is known so that the code results of U can be compared to the experimental results $(U_0-U)/u^*$.

From equation (3.3), it is also clear that the pressure P is an exact linear function of downstream distance L for an incompressible flow ($\rho \equiv$ constant). For a compressible flow for which $\langle U\rangle_A$ and ρ do not vary appreciably over L then equation (3.3) is still approximately true, and (3.3) can be written:

$$P = L\lambda\rho\langle U\rangle_A^2/4D + \text{constant} \tag{3.6A}$$

By applying the same argument which yielded equation (3.4), but now to a section of fluid centered on the channel centerline which is of arbitrary width (rather than of width D), then it is clear that the time averaged shear stress τ is a linear function of distance from the centerline. From

this it is also found that τ is also symmetrical about the centerline and is zero on the centerline. Written for the lower half channel then [7, p. 717]:

$$\tau/\tau_0 = 1 - 2y/D \qquad (3.7)$$

where y is the distance from the lower wall.

If the fluid equations are written in cartesian coordinates and time averaged as mentioned earlier, then the shear stress τ appears in the boundary layer approximation to the Reynolds averaged x-momentum equation as [1, p. 563]:

$$\rho(U\partial U/\partial x+V\partial U/\partial y) = -\partial P/\partial x + \partial\tau/\partial y \qquad (3.8)$$

where τ has the form [1, p. 563]:

$$\tau = \mu\partial U/\partial x - \rho\langle uv\rangle \qquad (3.9)$$

u and v are the instantaneous turbulent fluctuations of the total instantaneous velocity about the time averaged velocity, i.e. $v_x=U+u$, $v_y=V+v$, ($v_z=W+w$). The quantity $\langle uv\rangle$ is the time averaged value of uv, and $(-\rho\langle uv\rangle)$ is the familiar turbulent Reynolds shear stress [1, p. 563]. $(-\rho\langle uv\rangle)$ is not a true stress but is rather a momentum exchange effect. That $\langle uv\rangle$ is not zero is a well known phenomena of turbulent shear layers. A somewhat oversimplified explanation of this fact can be given by considering figure 3.3. It is seen that if a portion of the fluid in a shear layer moves to a region where U is higher, as a result $v>0$, then once it arrives at its new position it will have created a $u<0$ in the flow there. Here the oversimplying assumption must be made that the velocity of the section of fluid that moved upward was preserved as it

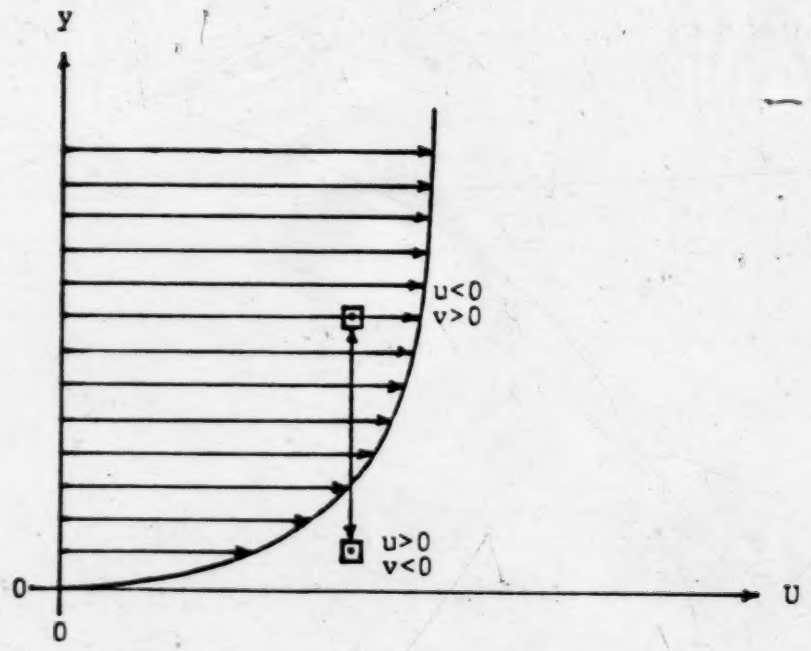

Figure 3.3 In a turbulent shear flow along a wall <uv> is less than zero. See page 36 for an explanation.

moved the small cross stream distance to its new position. For $v<0$ then $u>0$ will be created, so that on average uv is less than zero. The viscous part of τ in equation (3.9) is the usual viscous shear stress associated with molecular viscosity effects, and is experimentally found to be negligible in comparison with the Reynolds shear stress ($-\rho\langle uv\rangle$) when beyond a short distance from the wall.

Since by definition $\langle u\rangle=\langle v\rangle=0$, then it is necessary to define the quantities $\langle v^2\rangle^{1/2}/U_o$ and $\langle u^2\rangle^{1/2}/U_o$ to obtain a measure of the intensity of turbulence. The intensity of turbulence varies across the channel, however a typical value of $\langle u^2\rangle^{1/2}/U_o$ for channel flow is 5-10%. $\langle v^2\rangle^{1/2}/U_o$ is somewhat less than this, being about 75% of the x-direction value [11, p. 639, p. 725]. A better estimate for $\langle u^2\rangle$ can be obtained using figure 7-60 from [11], from which one can estimate that $\langle q^2\rangle/(u^*)^2 \cong 4$, where $\langle q^2\rangle=\langle u^2\rangle+\langle v^2\rangle+\langle w^2\rangle$; but, $\langle v^2\rangle\cong\langle w^2\rangle\cong 0.75\langle u^2\rangle$, so this yields $\langle u^2\rangle \cong 1.6(u^*)^2$. Substituting in equation (3.5) for u^*, then we have:

$$\langle u^2\rangle^{1/2}/\langle U\rangle_* \cong 1.26(\lambda/8)^{1/2} \qquad (3.9A)$$

3.2 Definition of layers in channel flow

For convenience, the channel is usually divided into several regions based on distance from the wall. The region closest to the wall, where the flow is directly affected by the condition at the wall, is called the "inner" or "wall"

region [11, p. 588]. Beyond this region towards the center, the flow is only indirectly affected by the wall and is called the core region. Very close to a perfectly smooth wall there is also a "viscous sublayer" and a "buffer" layer. The viscous sublayer is a very thin layer where the flow is predominantly viscous, between the wall itself and the inner region. The buffer region is the name given to the transition region between the viscous sublayer and the inner region. It can be shown [11, p. 718, p. 615] that the velocity profile is a linear function of y in the viscous sublayer, U being zero for y=0. It is also known that in the viscous sublayer $\mu\partial U/\partial y >> (-\rho\langle uv\rangle)$, so τ is essentially due to viscosity here. In contrast, in the core region the Reynolds stress is very much larger than the viscous shear stress. Throughout the buffer and inner layers, τ is due to both the viscous and turbulent shear stresses.

Defining the normalized cross stream distance $y^+ = yu^*/\nu$, then the viscous sublayer generally extends from the wall to $y^+ \sim 5$. In the case of flow past a flat plate, or high Reynolds number pipe and channel flow, the inner and buffer regions intersect at $y^+ \sim 30$ and the inner wall and outer regions usually overlap near $y^+ \sim 1000$; however, for low Reynolds number flow the channel half-width may be less than these distances. Since the inner region usually only occupies one-tenth or so of the channel, then in this case the distance y^+ from the wall to the boundaries between these regions must depend on the flow parameters.

3.3 Eddy size analysis

In modelling turbulent fluid flow on a computational grid, it is necessary to have the cell sizes about the same size as the smallest eddies of interest. We chose to simulate most of the range of eddy sizes in the channel, so that the approximate size of the smaller eddies was required. This size is known to be the Taylor microscale λ_z, defined by $(\lambda_z)^2 \equiv 2\langle u^2\rangle / \langle (\partial u/\partial y)^2\rangle$. Note that it is conventional in the literature to use a subscript g for the Taylor microscale, rather than a subscript z; however, due to word processing limitations, a subscript z had to be used. λ_z is a measure of the average dimension of the smallest eddies in the flow field [11, p. 41]. In order to determine the size of λ_z in terms of the channel width D, it is necessary to introduce the equation for the time averaged turbulence kinetic energy. This equation can be derived, with much work, by multiplying equation (2.2) by $\underline{v}$ and time averaging, and then subtracting from this the equation obtained by multiplying the time average of equation (2.2) by the time averaged velocity $\langle \underline{v} \rangle$ (not to be confused with $\langle \underline{v} \rangle^n$ from equation (2.15)). See Hinze [11, pp. 68-72] for a derivation in cartesian coordinates. The result of this is the equation for the change in kinetic energy of turbulence per unit mass per unit time:

$$0.5\partial(\langle q^2\rangle)/\partial t = -\nabla\cdot\langle \underline{v}'(p'/\rho + q^2/2)\rangle + \nabla\cdot[2\nu\langle \underline{\check{s}}\cdot\underline{v}'\rangle] - 2\nu\langle \underline{\check{s}}:\nabla\underline{v}'\rangle - \langle \underline{v}'\underline{v}'\rangle:\nabla\underline{V} \qquad (3.10)$$

where $\langle q^2\rangle = \langle u^2\rangle + \langle v^2\rangle + \langle w^2\rangle$ and the instantaneous velocity is

$\underline{v}=\underline{V}+\underline{v}'$ (note: $<\underline{v}>=\underline{V}$ and $\underline{v}'=(u,v,w)$); $\check{\underline{s}}$ is the fluctuating stress tensor $\check{\underline{s}} = (\nabla\underline{v}' + (\nabla\underline{v}')')/2$; and p' is the fluctuating component of pressure i.e. $p = P + p'$ (note: $<p>=P$). Despite their formidable appearance, each of the terms in equation (3.10) can be associated with a physical process [11, p. 72]. According to Hinze [11, p. 72], the last term in the equation is the turbulence production term, and the second last term is the rate of loss of $<q^2>$ to heat due to viscous dissipation. For a flow in which all time averaged quantities are constant in time (stationary turbulence), we can make the local equilibrium assumption that the production of turbulence is equal to the dissipation of turbulence [11, p. 75]. In a channel, the velocity gradient $\partial U/\partial y$ is dominant so the production p can be written [14]:

$$p \cong -<uv>\partial U/\partial y \qquad (3.11)$$

The viscous dissipation is largely due to eddies which have size λ_z or smaller, and these eddies are formed from much larger eddies. By the time the eddies have reached the size λ_z they are no longer dependent on the mean flow conditions and so are essentially locally isotropic. Thus, because most dissipation occurs at scales that are locally isotropic, we can use the exact expression for isotropic turbulence for the dissipation ϵ [11, p. 219]:

$$\epsilon = 15\nu<u^2>/\lambda_z^2 \qquad (3.12)$$

By defining ℓ as the average size of the energy containing eddies (i.e. the largest eddies in the flow), then ϵ is

governed by the supply of turbulence kinetic energy $\langle u^2\rangle/2$ and the time scale $\ell/\langle u^2\rangle^{1/2}$ of the large eddies. Taking ϵ as the ratio of these two quantities, then [11, p. 225]:

$$\epsilon = A\langle u^2\rangle^{3/2}/\ell \qquad (3.13)$$

where A is a constant near unity. We can also estimate the production $\mathcal{P}$ using a boundary layer approximation for equation (3.11) since $\partial U/\partial y \sim U_0 2/D$, and $u \sim v \sim \langle u^2\rangle^{1/2}$:

$$\mathcal{P} \sim \langle u^2\rangle U_0/\delta \qquad (3.14)$$

where $\delta=D/2$. Using our local equilibrium assumption we can equate $\mathcal{P}$ from equation (3.14) with ϵ from equation(3.13), so that we obtain (as in [14]):

$$\ell/\delta \sim \langle u^2\rangle^{1/2}/U_0 \qquad (3.15)$$

Equating equation (3.12) to (3.13) we also have that:

$$\ell/\lambda_z = 0.258A^{1/2}(\langle u^2\rangle^{1/2}\ell/\nu)^{1/2} \qquad (3.16)$$

By using equations (3.15) and (3.16) we have the desired result for λ_z in terms of known quantities:

$$(D/2)/\lambda_z \cong 0.23(U_0 D/2\nu)^{1/2} \qquad (3.17)$$

where a value of A=0.8 has been used in equation (3.13) (see [11, p. 248, p. 255]). Note that equation (3.17) was derived under the assumptions of local equilibrium of production and dissipation and for turbulence which is isotropic at the scale λ_z.

One can also define a third eddy size, η, which is associated with the size of the eddies which are on the verge of being completely dissipated. This is the smallest eddy size and is called the Kolmogoroff scale. η satisfies [11,p.225]:

$$\lambda_z/\eta = (15)^{1/4}(\langle u^2\rangle^{1/2}\lambda_z/\nu)^{1/2}$$

3.4 Analysis of distance to reach steady state velocity profile

In modelling channel flow numerically, the initial velocity condition of the channel was chosen for simplicity to be a linear profile i.e. the along stream velocity was initially set to be a linear function of distance from the wall throughout the entire channel. Before attempting to simulate the channel, it is thus necessary to develop a rough estimate for the time required for this velocity profile to transform into a curve resembling the fully developed profile. For simplicity, a constant velocity profile not a linear one is assumed in the following derivation. Note that a linear velocity profile fit to the experimental data would obviously approach the experimental profile quicker than that predicted by the following constant velocity profile derivation.

Assume in the computational grid cell nearest the wall that the viscous shear stress term $\mu\partial^2U/\partial y^2$ is about equal to the Reynolds stress term in the x-momentum time averaged equation

$$\rho dU/dt = -\partial P/\partial x + \mu\partial^2U/\partial x^2 + \mu\partial^2U/\partial y^2 - \rho\partial(\langle uv\rangle)/\partial y \qquad (3.18)$$

Note that $\partial U/\partial t \equiv 0$ so that it can be included in the equation as part of the Lagrangian derivative without any trouble, as

long as dU/dt is always interpreted as $dU/dt=U\partial U/\partial x + V\partial U/\partial y$. Using a boundary layer approximation, then certainly $\partial^2U/\partial x^2 << 2\partial^2U/\partial y^2$. In order to determine when $\partial P/\partial x << 2\mu\partial^2U/\partial y^2$ let us say that U in the first cell satisfies $U \sim U_o/2$ and U_o is about the same as $\langle U\rangle_*$, the average value of U across the stream; then the condition that $\partial P/\partial x << 2\mu\partial^2U/\partial y^2$ is:

$$\lambda\rho\langle U\rangle_*^2/4D << 2\mu U/\Delta^2$$

where equation (3.6A) has been used to obtain $\partial P/\partial x$ and a boundary layer approximation has been used for $\partial^2U/\partial y^2$ with the width of a cell Δ as the length scale. Since $U \sim U_o/2 \sim \langle U\rangle_*/2$, then this can be written as

$$\lambda\Delta/D << 2\mu/U\rho\Delta$$

By introducing the dimensionless cell width $\Delta^+\equiv(u^*)\Delta/\nu$, then equation (3.5) can be written:

$$U\rho\Delta/\mu = \Delta^+(2/\lambda)^{1/2}$$

so that the condition for $\partial P/\partial x << 2\mu\partial^2U/\partial y^2$ is:

$$\Delta^+ << [(\lambda/2)^{1/2}]D/\Delta \qquad (3.19)$$

λ is always less than 0.04 for turbulent flow [1, p. 613], and $D/\Delta \sim 30$ (i.e. the channel is 30 grid cells wide), so that equation (3.19) will be satisfied for $\Delta^+<<200$ or so, i.e. the first cell should not extend much further than the buffer layer so that $\Delta^+\leq 30$. With these assumptions equation (3.18) can be written for the cell next to the wall as:

$$\rho dU/dt \cong 2\mu\partial^2U/\partial y^2 \qquad (3.20)$$

This equation is valid throughout the first cell and so can be integrated over the volume of the cell. The right side

can be expressed as a surface integral since $\int\nabla^2 U dV = \int\nabla\cdot(\nabla U)dV = \int(\nabla U)\cdot\underline{dS}$ so that we can write, for a uniform grid spacing and assuming $\partial U/\partial x << \partial U/\partial y$:

$$\rho\int(dU/dt)dV \cong \mu(-\partial U/\partial y|_{y=0} + \partial U/\partial y|_{y=y_1})\Delta \qquad (3.21)$$

where y1 is the y position of the point midway between the first and second cells from the wall. If we now approximate the left hand side of (3.21) by assuming that dU/dt is constant at dU_1/dt over the cell, as it would be in a finite difference approximation, then $\rho\int(dU/dt)dV \cong \rho\Delta^2 dU_1/dt$, where Δ^2 is the volume of a cell of unit depth. If we also assume that U has the value $(U_0+U_1)/2$ in the second cell and approximate the derivatives of the right hand side of equation (3.21) with finite differences, then we have:

$$\rho\Delta^2 dU_1/dt \cong \mu(-5U_1 + U_0) \qquad (3.22)$$

Now $dU/dt=\partial U/\partial t+U\partial U/\partial x+V\partial U/\partial y$; but V=0 for a channel, and $\partial U/\partial t=0$ by definition, so: $dU/dt=U\partial U/\partial x$. Making the substitution $\xi=x/U$, then $dU/dt=\partial U/\partial\xi$ and the solution to equation (3.22) is:

$$U_1(\xi) = U_0(4\exp(-5\mu\xi/\rho\Delta^2) + 1)/5 \qquad (3.23)$$

The decay constant Γ for $U_1(\xi)$ is thus:

$$\Gamma = \rho\Delta^2/5\mu \qquad (3.24)$$

and U_1 will have reached a value $0.5U_0$ when $\xi=\Gamma$. Thus, the downstream distance the fluid must travel before the value of the velocity in the finite difference cell next to the wall changes from its initial value of U_0 to the value $0.5U_0$, because of viscous drag from the wall, is: $x_1=0.5U_0\Gamma$ i.e.

$$x_t = U_0 \rho \Delta^2 / 10\mu \qquad (3.25)$$

This provides us with a rough estimate of what distance we would expect a constant velocity profile to take to develop into a reasonably sheared velocity profile and what this distance x_t depends on.

CHAPTER IV

Numerical Simulation of Channel Flow

The full compressible fluid equations (2.2) and (2.6) written in two dimensional cartesian coordinates x and y are listed in Appendix A. These were modelled on the University of Alberta's Amdahl 5860 computer using a fluid particle approach.

4.1 Boundary and initial conditions

The channel geometry simulated was that of a fluid entering a channel with a linear velocity profile, as mentioned in section 3.4. Obviously, transition to a fully turbulent velocity profile could not be expected in the short length channels which computation times demanded. A typical ratio of length to width for a computational run was four, whereas one would not expect steady state flow for much longer lengths [11, p. 709]. However, by the addition of a random component to the entrance fluid velocities, enhanced transition to turbulence can be obtained [11, p. 600].

As an initial condition, the entire channel was given a linear velocity profile i.e. the alongstream velocity was a linearly increasing function of distance from the wall. The particles were given velocities which yielded a Gaussian velocity distribution about the desired initial linear velocity profile. This provided an initial flow that had a degree of random motion. A standard deviation of $\langle U \rangle_{\infty}/10$

was used for the Gaussian distribution, where <U>, is the average value of U across the channel. The fluid entering the channel was also given a linear velocity profile with a small random component added. The particles were all given the same initial temperature T_o and the number of particles per cell was a constant n_o, so that the pressure and density was initially constant everywhere. The particles were placed randomly within each cell.

Since the channel is symmetric about its centerline, only one half of the channel was modelled. The channel geometry and ghost points are depicted in figure 4.1. The x and y axes are offset from their conventional position for convenience of defining which cell a particle is in. On the line $y=D/2$ ($j=N_y+2$) the ghost points were given the values $(\underline{v}_x)_{i,j}=(\underline{v}_x)_{i,j-1}$, $(\underline{v}_y)_{i,j}=-(\underline{v}_y)_{i,j-1}$, $p_{i,j}=p_{i,j-1}$, $T_{i,j}=T_{i,j-1}$ because the full channel's centerline lies at $y = D/2 - \Delta/2$. On the lower boundary $\underline{v}$ must be zero to satisfy the usual no slip condition at the wall. As well, $T=T_o$, and $\partial p/\partial y=0$ here, so that the ghost points on the line $y=-\Delta$ must satisfy $\underline{v}_{i,j}=-\underline{v}_{i,j+1}$, $T_{i,j}=2T_o-T_{i,j+1}$ and $p_{i,j}=p_{i,j+1}$, where $j=1$. The particles are all specularly reflected from the $y=-\Delta/2$ and $y = D/2 -\Delta/2$ boundaries. The ghost points on the line $x=-\Delta$ were given values which yielded Neumann boundary conditions for $\underline{v}$, T and p i.e. $\underline{v}_{i,j}=\underline{v}_{i+1,j}$, $T_{i,j}=T_{i+1,j}$, $p_{i,j}=p_{i+1,j}$, where $i=1$. Similarly, Neumann conditions were used for $\underline{v}$ and T for ghost points on $x=L$; however, for pressure at $x=L$ it was

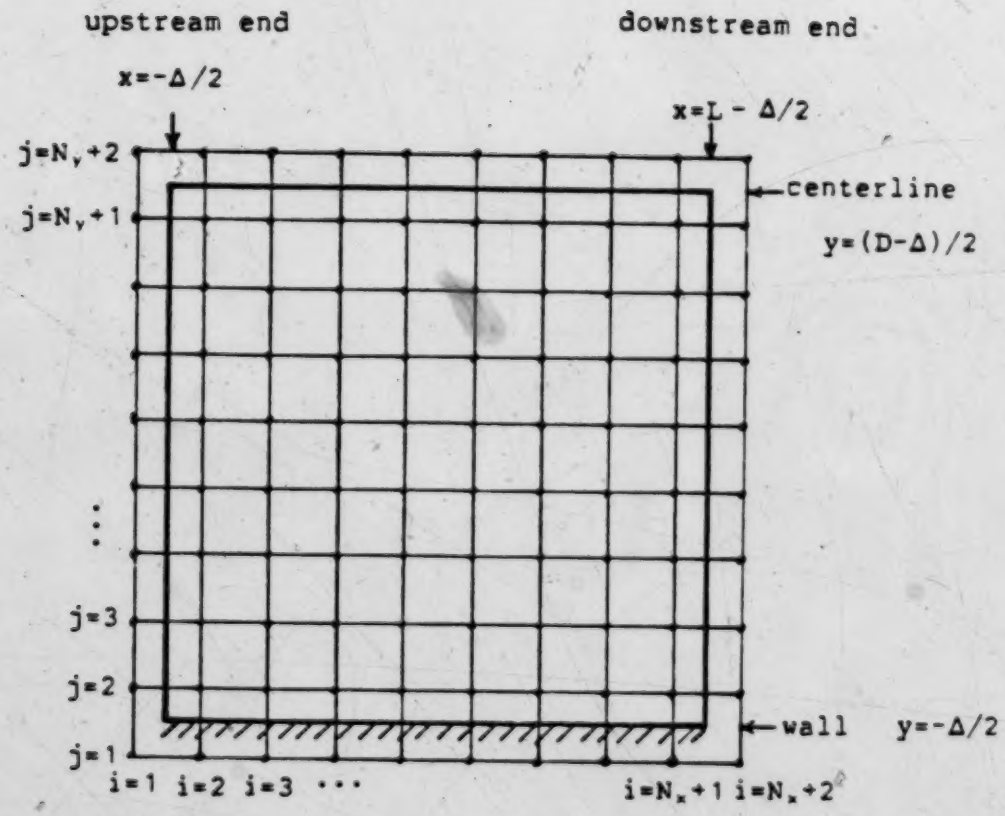

Figure 4.1 Ghost point configuration for channel simulation.

found that special consideration had to be given, as described in the following section.

4.2 Open boundary considerations for channel flow

4.2.1 Streaming instability

At the downstream boundary of the computational window a growing instability in the flow arose unless proper care was taken in applying the pressure boundary condition here. This instability has a very physical explanation; if the average number of particles in a cell is n_0, then after a particle crosses the downstream boundary of the computational window, the cell i,j which the particle left will have a lower pressure in general, since it has only $n_{i,j}=n_0-1$ particles per cell. Thus, the particles whose x-positions lie in the last two cells of the channel will feel a positive x-direction acceleration from the $-\partial p/\partial x$ force. The particles thus reach the end of the channel with a higher velocity than previous particles. If a Neumann condition is applied to the pressure at the boundary here, then the $(-\partial p/\partial x)$ assigned to particles in the last two cells of the channel varies linearly between zero and the positive value of $(p_{i-1,j}-p_{i,j})/\Delta$ (using a modified ∇p differencing with $p_{i-2,j}=p_{i-1,j}$ and $p_{i-1,j}>p_{i,j}$). However, particles are now leaving the right hand side of cell i,j at a faster rate than they enter the left hand side, since particles are accelerated across the last two cells. Thus

$n_{i,j}$ will drop faster than $n_{i-1,j}$, producing a larger $(-\partial p/\partial x)$ so that the particles leave the channel with an even higher velocity. One can easily see that this particle accelerating phenomena will work its way upstream, resulting in a completely nonphysical growing loss of particles out the downstream boundary. This undesirable situation can be remedied by "clipping" the pressure in the last few cells of the window. By not allowing the pressure to drop below a certain specified value in these cells, the above instability cannot develop. That is, if the density and temperature are such that the actual pressure of the cell is less than the clipping value, then for all finite differencing the clipped value was used instead of the actual pressure. It was convenient to use equation (3.3) to obtain an estimate of the pressure drop expected across a length of channel. A pressure 3.5% of p_o less than that predicted from (3.3) was then applied as the clipping value in order to minimize unnecessary clipping of turbulent pressure variations. To ensure that the instability didn't develop, clipping was done in the second last cell as well, but with a lower minimum pressure so that the mininum permissable pressure in the second last cell was set at about 2% of p_o less than that of the last cell.

4.2.2 Buffer zones

The application of pressure clipping removes the growth of the above instability, however there can still be a

nonphysical effect resulting from particles crossing the window boundaries. At the downstream boundary this effect is the sudden appearance of a false pressure drop in the cell the particle left (as long as $p>p_c$ where p_c is the clipping pressure). If at the upstream boundary, then the pressure will rise suddenly in a cell where a particle is introduced into the window. Both of these anomalies are a result of suddenly ignoring a particle's contribution to the densities in the cells near the boundary once a particle is beyond the window. The solution to this problem was the addition of buffer zones at each end of the channel. Particles in the buffer zones were left to coast with no fluid forces acting on them. Particles were created at the upstream boundary of the upstream buffer. Once a particle crosses the downstream side of this buffer it enters the window carrying the entrance velocity and temperature conditions. However, at the same time, another particle is created at the upstream side of the buffer so that in a short time, after this particle crosses the width of the upstream buffer, it too will enter the window. The random component of the entrance velocity, mentioned earlier, was obtained by giving the upstream buffer particles the Gaussian velocity distribution (which was also applied as an initial condition to the window particle velocities). The upstream buffer particles were also given a linear velocity profile, in order to maintain the linear entrance velocity profile.

When a particle crosses the downstream boundary it is placed in the downstream buffer where it coasts until it crosses the downstream boundary of the downstream buffer, after which it is dropped from memory. The density of the cells in the channel are then dependent on the value of $n_{i,j}$ at grid points inside the buffers. In this way particles do not cause artificial shocks as they exit and enter the window but are treated in the same manner as a particle inside the window crossing an interior cell interface.

It is also convenient to calculate the value of the density in the cells of the downstream buffer which are adjacent to the window. One can then calculate the pressure in these cells and use these values as the boundary condition pressures for the last cells of the window. For simplicity, the value of $\langle T \rangle^n$, from the temperature equivalent of equation (2.18), was used for the temperature in the buffer cells in which pressure ($p=\rho RT$) was calculated, rather than calculating $T^{(n)}$ from the temperature version of equation (2.15).

Both buffers were given an initially constant value (T_0) for the particle temperatures and initially contained n_0 particles per cell. Both buffers were made long enough so that a particle at the ends furthest from the window boundary of either buffer would contribute an amount just within machine precision to the density of cells inside the window at the window-buffer boundaries. For a Gaussian mass density profile of width $a^2=\Delta^2/\sqrt{2}$, this required buffers

about four cells long for single precision on the Amdahl 5860. The two buffers were also given Gaussian particle velocities with standard deviation 1/10 the initial total average fluid velocity.

4.3 Other considerations

The necessity of using the ∇p differencing explained in Chapter 2 was mentioned there. It should be noted that using the modified ∇p differencing, described in Chapter 2, yielded values of $\langle u^2 \rangle^{1/2}/U_0$ which were as small as one tenth the values obtained using conventional ∇p differencing. This is a significant improvement for the NGP approach, putting it on the same level, in terms of random particle noise ("thermal heating"), as the SUD grid particle assignment and area weighting methods described in [7].

4.3.1 Gaussian averaging of v and T

It was found that in order to best obtain a smooth fluid picture from the particles, a one dimensional averaging in the x-direction was desirable. Previous NGP codes [2] have used an arithmetic averaging over three cells; this, however, reduces resolution considerably. To obtain a local smoothing without a significant reduction in the effective grid size, a one dimensional density weighted Gaussian averaging with width $a^2=\Delta^2/\sqrt{2}$ was used on velocity and temperature. That is, the values of velocity and temperature from equation (2.15) (and it's temperature

equivalent) were averaged in the x-direction with a Gaussian weighting before being used in the finite differencing to obtain $\underline{L}$ and E in equations (2.16) and (2.17). The form of the Gaussian averaging written for the fluid velocity at $\underline{r}_{i,j}$ at time $t=n\Delta t$ is:

$$(\hat{\underline{v}}^{(n)})_{i,j} = [\Sigma_{m\neq m0} f(\underline{r}_{i,j}-\underline{r}_{m,j})(\underline{v}^{(n)})_{m,j}\rho^n_{m,j}]/ [\Sigma_{m\neq m0} f(\underline{r}_{i,j}-\underline{r}_{m,j})\rho^n_{m,j}] \qquad (4.1)$$

where f is defined in equation (2.8), and the sums are over all grid points in the jth row except those which have no particles in them, the m0 indicating these empty cells. Empty cells are excluded from the averaging because $\underline{v}^{(n)}$ and $T^{(n)}$ cannot be defined in an empty cell using an equation of the type given by (2.15). The value of $(\underline{v}^{(n)})_{m,j}$ is that from equation (2.15), and the value of averaged velocity $(\hat{\underline{v}}^{(n)})_{i,j}$ from (4.1) is then used in place of $(\underline{v}^{(n)})_{i,j}$ to obtain the finite differencing in the $\underline{L}$ and E of equations (2.13) and (2.14). For $a^2=\Delta^2/\sqrt{2}$ ($a\cong 0.84\Delta$), this averaging uses a factor $f(0)=1$; a factor $f(\Delta)\cong 0.49$ for cells adjacent to cell i,j; and a factor $f(2\Delta)\cong 0.06$ for cells two cells away, compared to $f(0)=f(\Delta)=1$ for a horizontal arithmetic averaging over adjacent cells. The averaging is thus not as large scale as the arithmetic case.

Without the Gaussian averaging, a few empty cells developed, while virtually none existed with the averaging. In addition, the averaging helped reduce the random noise caused by the NGP approximation mentioned in section 4.5. The removal of the Gaussian averaging for velocity and

temperature produced turbulent intensities about three times that obtained for $a^2=\Delta^2/\sqrt{2}$, in an early run with $\alpha_m=\alpha_t=0.1$ (α_m and α_t are from equations [2.16] and [2.17]; note this is too large a value of α_m for channel flow - see section 4.5)

It should be mentioned that in the computational versions of equations (4.1) and (2.7), only cells for which f was within machine precision were used in calculating the sums, since cells further away would have insignificant contributions. The subroutines for the calculation of density from equation (2.7), and temperature and velocity from equations of the form of (4.1), were done on a number crunching array processor (FPS 190L) attached to the Amdahl 5860.

4.3.2 Empty cells

With the modified ∇p differencing described in Chapter 2, very few empty cells were produced in any of the runs (using conventional ∇p differencing, one in ten cells was empty; while with the modified ∇p there were usually no empty cells). However, the particle approach described in this work, which obtains the fluid density based on the particle positions on a uniform grid, would not lend itself well to modelling a system with order of magnitude density variations, since this would require some cells to have far more particles than necessary for good accuracy, adding unnecessary expense to a run; while some cells would have

only a few or even no particles, resulting in poor resolution of the fluid here. A variable grid might be introduced, but this would add problems of its own. For those interested in modelling a system that might have empty cells, or "holes" as we have called them, it is critical to note that holes cause various problems in the conservation of fluid quantities when using flux conservative differencing. This is because there are no particles in the hole which can cancel the flux conservative differencing from neighboring cells. Consider, for example, the $(\partial^2 T/\partial x^2)_{i,j}=(T_{i+1,j}+T_{i-1,j}-2T_{i,j})/\Delta^2$ term appearing in E from equation (2.17). (Actually the temperature used here is that from the temperature equivalent of (4.1) but the circumflexes will be left off for convenience.) With no empty cells, then $(\partial^2 T/\partial x^2)_{i+1,j}=(T_{i+2,j}+T_{i,j}-2T_{i+1,j})/\Delta^2$ and $(\partial^2 T/\partial x^2)_{i-1,j}=(T_{i,j}+T_{i-2,j}-2T_{i-1,j})/\Delta^2$, so that the sum on the right hand side of equation (2.21) would properly vanish. However, if cell i,j is empty, then there are no particles which will yield a value of $(\partial^2 T/\partial x^2)_{i,j}$ in the inner sum of the temperature equivalent of equation (2.19). Thus, the $-2T_{i,j}$ term in $(\partial^2 T/\partial x^2)_{i,j}$ (note that with Gaussian averaging, the fluid temperature, velocity and density are defined even in an empty cell) will not cancel the $T_{i,j}$ term of $(\partial^2 T/\partial x^2)_{i+1,j}$ and the $T_{i,j}$ term of $(\partial^2 T/\partial x^2)_{i-1,j}$, so that the right hand side of equation (2.21) is nonzero, and we no longer have energy conservation. This situation obviously occurs for all the

terms in both the momentum and energy equations which are differenced flux conservatively, and which should thus vanish when summed over the grid.

For all terms except the ∇p term of the momentum equation this situation can be partially remedied by applying a Neumann condition for the values of temperature, velocity, and pressure at the interface of the empty cell. For example, instead of using the value of $T_{i,j}$ in the expression for $(\partial^2 T/\partial x^2)_{i+1,j}$, one would use the value of $T_{i+1,j}$, and for $(\partial^2 T/\partial x^2)_{i-1,j}$ one would use $T_{i-1,j}$ in place of $T_{i,j}$. In this way all $\partial^2 A/\partial x^2$, $\partial^2 A/\partial y^2$ terms will vanish exactly when summed over the grid. Unfortunately, the single derivative terms and the mixed derivatives will not exactly vanish using this scheme, since their flux conservative finite difference forms are based on cancellation over cells which are two grid points apart. For example, $(\partial T/\partial x)_{i,j}=(T_{i+1,j}-T_{i-1,j})/2\Delta$ is normally cancelled by a $-T_{i+1,j}$ from $(\partial T/\partial x)_{i+2,j}$ and a $T_{i-1,j}$ from $(\partial T/\partial x)_{i-2,j}$. If cell i-2,j is empty then there will be no $T_{i-1,j}$ from $(\partial T/\partial x)_{i-2,j}$, but there will be an extra $-T_{i-1,j}$ by applying a Neumann condition for $(\partial T/\partial x)_{i-1,j}$. Similarly, we will accumulate $2T_{i-3,j}$ on the other side of the hole. If $2T_{i-3,j}=2T_{i-1,j}$, then the sum will vanish, but in general this will not be exactly true. This difficulty is illustrated schematically in figure 4.2. In the top left figure, a flux conservative finite differencing of $\partial^2 A/\partial x^2$ will vanish exactly when summed over the grid; similarly for

$\partial A/\partial x$ as in the top right figure. However, when a hole appears in cell i,j (as depicted by the shaded region), then a term $-A_{i-1,j}+2A_{i,j}-A_{i+1,j}$ (as depicted in the middle figure), will be present in a sum of $\partial^2 A/\partial x^2$ over the grid, so that the differencing is no longer conservative. By applying a Neumann condition at the empty cell interface, the problem is corrected, as in the bottom left picture. Unfortunately, $\partial A/\partial x$ remains nonconservative. In the middle right figure, we see an extra $A_{i-1,j}-A_{i+1,j}$ appears with no Neumann condition; while applying the Neumann condition gives an extra $2A_{i-1,j}-2A_{i+1,j}$ as in the bottom right figure. It is clear that it would not be wise to use a Neumann condition for evaluating ∇p in cells adjacent to a hole, since there would then be no tendency for the hole to be filled in. Note that over a long time period with steady state conditions one would expect the nonzero terms in the sums of $\underline{L}$ and E over the grid, which are caused by empty cells, to average to zero.

4.4 Constraints on variables

The quantities involved in the description of channel flow must satisfy various conditions before we would expect a simulation to converge to the correct solution. As mentioned in Chapter 1, the Courant (CFL) stability criterion must be satisfied, i.e.

$$\{|\underline{v}_i|_{mx} + \mu/\rho\Delta + (C_s^2+v_i^2)^{1/2}\}\Delta t/\Delta \leq 1 \qquad (4.2)$$

where $|\underline{v}_i|_{mx}$ is the maximum particle velocity; C_s is the

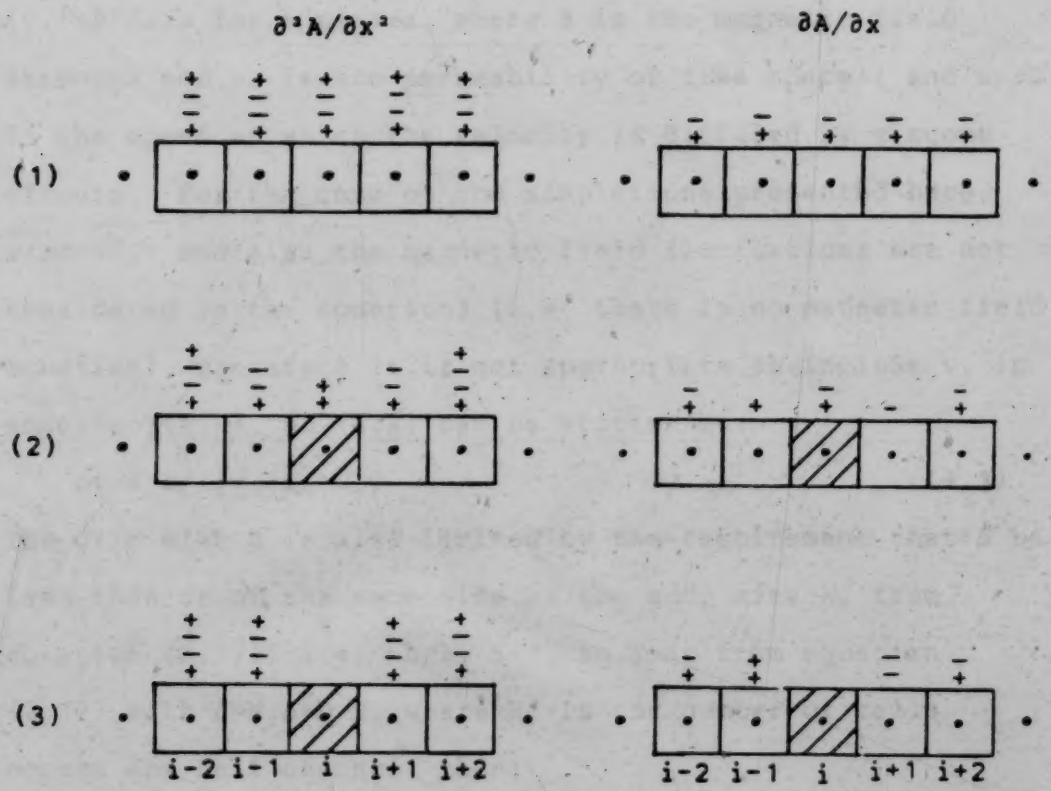

Figure 4.2 Cells with no particles in them introduce a nonconservative effect into the normally conservative flux conservative finite differencing. A plus/minus sign above a cell signifies that the value of A from that cell is added/subtracted to/from a sum of the differencing over the grid. (1) No empty cells, (2) empty cells - no corrective action in the differencing and (3) Neumann treatment of empty cells.See p. 59 for a further explanation.

adiabatic sound speed (for an ideal gas $C_s=[\gamma P/\rho]^{1/2}$ where γ is the ratio of specific heats); v_t is the Alfvén wave speed ($v_t^2=B^2/\mu_0\rho$ for a plasma, where B is the magnetic field strength and μ_0 is the permeability of free space); and $\mu/\rho\Delta$ is the speed at which the velocity is diffused by viscous effects. For the case of the simulations presented here, $\mu/\rho\Delta<<C_s$; and also the magnetic field fluctuations are not considered in the equations (i.e. there is no magnetic field equation), therefore it is not appropriate to include v_t in equation (4.2), so (4.2) can be written as:

$$\Delta t \leq \Delta/(|\underline{v}_i|_{mx}+C_s) \qquad (4.3)$$

The grid size Δ is also limited by the requirement that Δ be less than or of the same size as the eddy size λ_z from equation (3.17), i.e. $\Delta/\lambda_z \leq 1$, so that from equation (3.17) with $\delta=N_y\Delta=D/2$, where N_y is the number of cells across the half channel, then:

$$0.23(N_yU_0\rho\Delta/\mu)^{1/2}/N_y \leq 1 \qquad (4.4)$$

Since we are interested in modeling turbulent fluid flow, it is necessary to maintain the Reynolds number above the turbulent-laminar transition region. For $Re_d=<U>_s D/\nu<2300$, where $<U>_s$ is the mean velocity of the channel, the flow is nearly always laminar [1, p.39]. The change to turbulent flow occurs over a range of Reynolds numbers slightly above this. With very undisturbed inlet conditions, the transition to turbulence may not occur until $Re_d>10,000$; however, it was mentioned earlier that well disturbed inlet conditions were used. A Reynolds number of $Re_d=5333$ was

thus regarded as a safe lower limit, i.e.

$$\langle U\rangle_s D/\nu \geq 5333 \tag{4.5}$$

In order that the fluid develop a velocity profile that isn't too far from the steady state profile by the time it reaches the end of the window, we require that the window be at least about twice the distance x_t given in equation (3.25). With the length of the channel given by $L=N_x\Delta$ then this requirement is (letting $U_0\cong\langle U\rangle_s$):

$$N_x\Delta \geq \langle U\rangle_s \rho\Delta^2/5\mu \tag{4.6}$$

The time required for the fluid to cross the window and develop the correct velocity profile must also be within a reasonable number of time steps for expense reasons. If this number of time steps is NT_x, and Δt has been chosen as a constant $\Delta t=r\Delta/(C_s+\langle U\rangle_s)$, where $|r|\leq 1$; then (3.25) gives:

$$NT_x=(C_s+\langle U\rangle_s)\rho\Delta/5\mu r \tag{4.7}$$

Since the data of Nikuradse and Dönch (figure 3.2) is for incompressible flow, then the velocity of the flow must be low enough that compressible effects do not become significant. From [1, p. 10] this requirement is:

$$U_0 \leq 0.3C_s \tag{4.8}$$

where U_0 is the midstream velocity. Also, the length of the channel should be such that it is at least about ten large scale eddy lengths long in order to have reasonable turbulent modelling. From equation (3.15) with $\langle u^2\rangle^{1/2}/U_0 \sim 0.1$ as mentioned in Chapter 3, then this requires:

$$L/\delta \geq 1$$

i.e. the length to width ratio of the window should be greater than 1 or so.

To satisfy the requirements of equations (4.3) to (4.8), the following values of constants were chosen:

$r=0.2$, $C_s=374$ m/s, $N_y=16$, $N_x=100$, $\rho=1.0$ kg/m^3,

$\langle U\rangle_s=112.2$ m/s, $\mu=1.74169\times10^{-4}$ kg/ms, $\Delta=2.5875\times10^{-4}$ m.

This yields a Reynolds number of $Re_s=\langle U\rangle_s D/\nu=5333$. $\langle U\rangle_s$ is constant in time because the left buffer maintains a constant rate of flow into the computational window. One should realize that at this low Reynolds number the turbulence is probably not isotropic at the scale λ_z, since the large scale eddy size ℓ is about the size of the Taylor microscale, based on equation (3.16). As a result, eddies of size λ_z are likely still affected by the eddies of size ℓ, which are not isotropic since the large scale eddies are dependent on the mean flow. Equation (3.17) is thus only a very rough approximation for λ_z. In addition, the following other parameters were chosen as:

$p|_{t=0}=100{,}000$ Pa, $T_0=348$ K, $Cv=732$ J/kgK,

$\kappa=2.94\times10^{-2}$ J/smK

Also, the linear velocity profile was characterized by a velocity of 81 m/s at the wall and 155 m/s on the channel centerline. Note that with these values the dimensionless cell width Δ^+, from Chapter 3, is $\Delta^+\cong11$; thus, the first cell extends only into the buffer layer so that equation (3.19) is satisfied, and as a result equation (3.25) is valid. It was found that using $n_0=m^2$ particles per cell,

where m is an integer, was desirable for random placement of the particles. In this way each particle could be placed randomly in an area occupying $1/m^2$ of a cell. Values of m, as well as a_m and a_1 (from equations [2.16] and [2.17]) are discussed in the next two sections.

4.5 The artificial drag term and noise caused by a NGP approximation

It was initially thought that the values of a_m and a_1 used in a simulation (a_m and a_1 are the artificial drag coefficients in equations [2.16] and [2.17]) would not affect the results of the simulation in any significant way, since they contribute no net force to any cell. Of course, if too small values are used then there is a large amount of multistreaming and the simulation becomes too noisy to yield any realistic results. However, for the channel simulation, it was found necessary to have a_m such that the intracell artificial drag forces were of the same size as the intercell molecular viscous forces. It is not known exactly why this is necessary, but it is reasonable, since, in a sense, the artificial drag force models intracellular viscosity so that intracell viscosity (caused by the artificial drag term) should be the same as intercell viscosity (caused by the actual viscous terms appearing in the fluid equations). In a turbulent boundary layer flow where molecular viscous effects are expected to extend to subgrid length scales, using a multistreaming term that is

much larger than the dominant viscous term is not realistic and causes the effect of intercell viscous forces to be swamped by the effect of intracell drag forces, resulting in an unrealistic flow. However, in a laminar flow, large values of a_m could probably be used without any difficulty since, with reasonable grid resolution, there should be no subgrid velocity variations, so that the particle velocities should be very closely tied to the fluid velocity.

If too large a value for a_m was used in the turbulent channel simulation, various incorrect velocity profiles resulted. Although the temperature equation was used for stability in the channel code, the channel flow was not a thermal boundary layer so that the effect of large a_t on the flow was not destructive. However, in a flow where subgrid conduction effects exist, one would probably have to use a value of a_t which yielded intracell (artificial) conduction terms which were the same size as intercell (actual) conduction terms. In the gas discharge simulation, the temperature profile was almost completely determined by Joule heating of the fluid, so that using large values of a_t (a_t=0.5) was also not destructive. As well, for the discharge geometry, large values of a_m were acceptable, since there was no boundary layer flow in the r,z velocities and the θ velocity was essentially laminar because of the two dimensional nature of the code.

Let us now develop an estimate of the order of magnitude of actual viscous particle accelerations and of

particle accelerations caused by the multistreaming term.

The actual change in along stream velocity that a particle undergoes in one time step due to molecular viscous effects is: $(\mu\partial^2 v_x/\partial y^2)\Delta t/\rho$, which can be estimated as: $\mu\langle U\rangle_*\Delta t/\rho\Delta^2$. The change in a particle's velocity due to artificial drag can be estimated as follows. If we say that, at time $n\Delta t$, a particle's velocity, $\underline{v}_i^{n-1/2}$, differs from $\langle\underline{v}\rangle^n$ of equation (2.18) by $\langle U\rangle_*$, i.e. $(\underline{v}_i^{n-1/2} - \langle\underline{v}\rangle^n) \sim \langle U\rangle_*$, then the artificial drag term will change that particle's velocity by $a_m\langle U\rangle_*$. If we were to have the effects of the molecular viscous and artificial drag terms about equal per time step, then this would require that $a_m\langle U\rangle_* = \mu\langle U\rangle_*\Delta t/\rho\Delta^2$ so that:

$$a_m \sim \mu\Delta t/\rho\Delta^2 \qquad (4.9)$$

Note that the effect of the artificial drag term on a particular particle decreases geometrically in time, since the magnitude of the change in that particle's velocity caused by this term depends on $|\underline{v}_i^{n-1/2} - \underline{v}^n|$, which is decreasing in time because of the effect of the artificial drag term. Thus, it might be better to compare the two terms based on how many time steps they take to change a particle's velocity by, say, $\langle U\rangle_*/2$. For the molecular viscous term let this number of time steps be K, so that $K(\mu\langle U\rangle_*\Delta t/\rho\Delta^2) = \langle U\rangle_*/2$ i.e. $K = \rho\Delta^2/2\mu\Delta t$, or, using $\Delta t = r\Delta/(C_*+\langle U\rangle_*)$ as in obtaining equation (4.7), then:

$$K = \rho(C_*+\langle U\rangle_*)\Delta/2\mu r \qquad (4.10)$$

(note the similarity to 4.7). For the artificial drag term,

let us say that a particle's velocity is initially $\langle U \rangle_i$ above $\langle \underline{v} \rangle$"; then, the number of time steps, M, needed to reduce a particle's velocity by $\langle U \rangle_i/2$ is found from $(1-a_m)^M = 1/2$, so that:

$$M = -\ell n2/\ell n(1-a_m) \qquad (4.11)$$

In order to have K=M, then we must have a_m such that

$$a_m = 1 - 2^{-1/K} \qquad (4.12)$$

where K is given by (4.10). Equation (4.9) would suggest a value of $a_m \cong 0.0003$ using the parameters listed in section 4.4, while equation (4.12) would suggest a value of $a_m \cong 0.0004$. If values of a_m much larger than this were used, incorrect velocity profiles resulted, as mentioned earlier. For example, it was found that, although noise levels were neglible with a value of $a_m = a_t = 0.1$ with 9 particles per cell, an incorrect velocity profile resulted, as shown in figure (4.3). A value of $a_m = 0.01$, $n_o = 9$ also yielded an incorrect velocity profile. A value of $a_m = 0.0015$ was used successfully in the channel. Unfortunately, when values of a_m this small were used, the particle velocities took on random components that were as large as actual turbulent fluctuations. These random velocities resulted largely because of random variations in pressure, which were in turn caused by random variations in density and temperature. The random fluctuations in density and temperature were a result of the discrete nature of a NGP fluid particle method. Using a value of $a_t = 0.5$ kept temperature fluctuations to low levels and also helped reduce the particle velocity noise

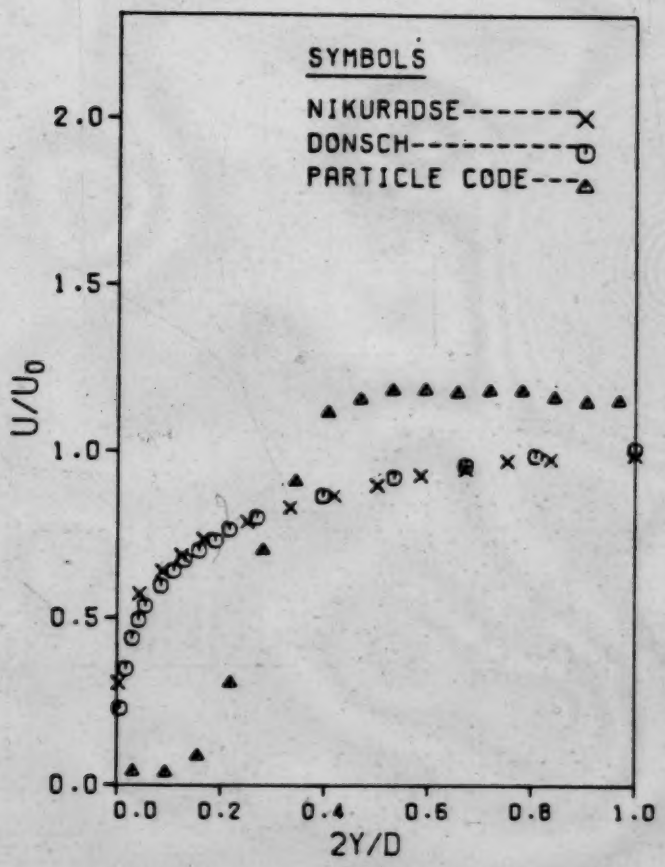

Figure 4.3 A velocity profile that results for channel flow if too large a value of the artificial drag coefficient a_m is used. The code results are indicated by the triangles, while the experimental results of Nikuradse and Dönch [12, p. 482] are indicated by X and 0 respectively.

levels, however a way of reducing the random density fluctuations needs to be developed; that is, if using more particles per cell is not an acceptable means of reducing noise. These random density variations result largely from particles flowing across cell boundaries. As the particles flow across the grid, the number of particles per cell $n_{i,j}$ fluctuates randomly because there are a finite number of particles randomly placed in a cell. Using a NGP grid particle assignment, this results in random density variations which in turn, through the ∇p force, yields noise in the particle velocities.

To quantify this effect of the NGP approximation on noise levels, consider a simple one dimensional case where all cells have m^2 particles in them (m is an integer), except cell i+1,j, which has $n_{i+1,j} = m^2-\delta$ particles in it. Here, the fluctuation in n is due simply to a finite number of randomly placed particles flowing across a grid; it is not a fluctuation due to actual fluid turbulence. With a value of $a^2=\Delta/\sqrt{2}$, then $f\cong 0.5$ in adjacent cells and $f\cong 0$ for other cells, so that, using equation (2.7):
$\rho_{i,j} \cong \rho_0(1 - \delta/4m^2)$, $\rho_{i+1,j} \cong \rho_0(1 - 2\delta/4m^2)$. Thus, between cells i+1,j and i,j a ∇p force

$$\nabla p \cong p_0\delta/4m^2\Delta \qquad (4.13)$$

develops (ignoring the effects of a modified ∇p differencing) due to the fluctuation in n. In one time step, the change in a particles's velocity due to this false ∇p force is:

$$\Delta v = \nabla p \Delta t/\rho_o \qquad (4.14)$$

If the velocity of the fluid in cell i,j is U and if it takes x steps for a particle flowing at speed U to cross a cell ($x=\Delta/U\Delta t$), then it will take about $x\delta/m^2$ time steps before this density gradient is corrected by other particles flowing into the cell i+1,j from cell i,j. This is because there are δ particles in a δ/m^2 area fraction of the cell, so that δ particles should flow into cell i+1,j in $x\delta/m^2$ time steps (assuming square cells). In two dimensions, there are three other cells adjacent to cell i+1,j from which particles could also flow into cell i+1,j, but let us assume that most of the particles move into cell i+1,j from the upstream cell, i,j. Using equations (4.13) and (4.14), then in $x\delta/m^2$ steps, the change in a particle's velocity would be:

$$x\delta\Delta v/m^2 = p_o\delta^2/4\rho_o U m^4$$

which creates a turbulent intensity of order:

$$u'/\langle U\rangle_i = p_o\delta^2/4\rho_o U m^4 \langle U\rangle_i \qquad (4.15)$$

where $u' \equiv x\delta\Delta v/m^2$. This suggests that the noise levels caused by particles crossing a boundary should be much less for runs with higher values of m^2. However, the above analysis fails to account for the fact that, for m^2 not too large, the value of δ would likely increase as m^2 increases (for large m^2 this would not be true since one would be approaching a continuous system), because the fluctuation across the cell boundaries is a result of the random placement of particles. With more particles randomly placed

in a cell, δ would be larger, although the percentage fluctuation in the number of particles per cell would be less. If we say that $\delta \sim m$, (4.15) becomes:

$$u'/\langle U\rangle_1 \sim p_o/4\rho_o U m^2 \langle U\rangle_1 \qquad (4.16)$$

Using the values of p_o, ρ_o, $\langle U\rangle_1$ listed earlier and assuming $U=\langle U\rangle_1$, then this yields noise levels of $u'/\langle U\rangle_1 = 0.23$, 0.12, 0.08 for $m^2 = 9$, 16, 25 respectively. Equation (3.9A) yields an analytic turbulence intensity of $\langle u^2\rangle^{1/2}/\langle U\rangle_1 \cong 0.09$ so that the noise levels are of the same order as the turbulent intensities. Computationally, it was found that approximate values of $\langle u^2\rangle^{1/2}/\langle U\rangle_1$ with $a_1=a_m=0.0015$ were: 0.35, 0.26, 0.25 for $n_o=9$, 16, 25; subtracting the turbulent intensity of 0.09 from these values, then noise levels of 0.26, 0.17 and 0.16 were observed for $n_o=9$, 16, 25 respectively. Comparing the values of 0.26, 0.17 and 0.16 from the code to the values of 0.23, 0.12, 0.08 from equation (4.16), it appears that the noise level due to random density fluctuations does not decrease quite as rapidly as $1/m^2$ with increasing m^2 as (4.16) predicts, although equation (4.16) is only an order of magnitude estimate. Note also that we have not accounted for the noise caused by temperature variations mentioned earlier. Since temperature variations appear directly as pressure variations, one might expect the noise caused by such variations to be of the same order as the density fluctuation noise levels. Increasing a_1 to 0.5 removed most of the temperature fluctuations and thus yielded less particle velocity noise. With $a_1=0.5$ and

a_m=0.0015, runs with n_0=9 and 25 gave values of $\langle u^2\rangle^{1/2}/\langle U_t\rangle \cong 0.22$ and 0.16 respectively, considerably lower than the values of 0.35 and 0.25 found with a_t=0.0015.

Because the particle noise is largely random, its effect on time averages like U and P can average out so that good results can be obtained for these quantities, even with low values of m. Physically, however, if too much random mixing occurs across the channel, then the velocity profile is diffused into a constant value across the channel. This was in fact observed in earlier runs which used the values $a_m=a_t$=0.0015. For quantities like $\langle u^2\rangle$ and $\langle uv\rangle$, noisy results are obtained since the particle noise is inseparable from turbulence in these averages and so the noise does not average out.

Although the particle noise caused by a NGP approximation interferes with obtaining values for turbulent correlations like $\langle u^2\rangle$ and $\langle uv\rangle$, the noise does not have a large effect on the fluid dynamics, as long as noise levels are reasonable. This is clear when one considers that with a_t=0.5, the temperature fluctuations are small so that the major source of the noise is a random difference in the number of particles in a cell. These random differences last only a fraction of the time required for a particle to cross a cell, as mentioned in deriving equation (4.15). As well, these random differences are likely overcompensated for, because of the lag time between the random accelerations and the random velocities. That is, if

particles in an adjacent cell are randomly accelerated toward a cell with fewer particles, they will maintain a higher velocity in that direction even after the acceleration stops, resulting in too many particles entering the cell; the particles are then accelerated in the opposite direction - the result is an oscillation of particles between adjacent cells, which does not contribute much to the fluid motion.

4.6 Results

With these considerations made, good results were obtained. A comparison of the time averaged along stream velocity with experimental data is shown in figure 4.4 for n_o=25, a_m=0.0015, a_1=0.5. The data of Nikuradse and Dönch as it appears in figure 3.2 was unnormalized as described in section 3.1. The code results were obtained from a time average over the last 100 time steps and a lengthwise spatial average over the section x=0.8L to x=0.96L. With this averaging not all turbulence was removed from U, so that the code results oscillated by small amounts about the experimental data. This is seen in figure 4.4 near 2y/D=0.3 where the code results are a bit lower than the experimental data and at 2y/D=0.9 where the code results are a bit higher than the experimental data. These oscillations occurred at different values of 2y/D for different time steps. With 9 particles per cell, the velocity profile agrees well with experimental results by 600 time steps. Runs up to 1800

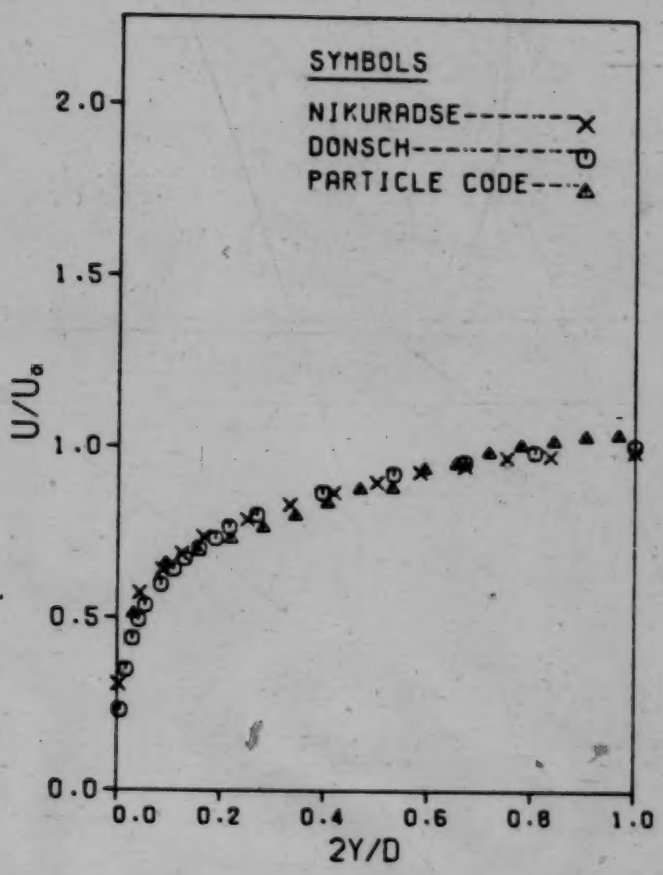

Figure 4.4 Comparison of code velocity profile (triangles) with experimental data of Nikuradse (x) and Dönch (0) from [12, p. 428] after 4200 time steps (4.47 ms).

time steps were done with 9 particles per cell; the code continued to maintain profiles as in figure 4.4 for the full length of these runs. With 25 particles per cell, random noise levels were lower so that turbulence required a longer time to develop, since turbulence development is enhanced by the random noise (as mentioned in section 4.1). As a result, runs with 25 particles per cell required a longer time to develop a steady state velocity profile. For n_o=25, steady state fully developed velocity profiles were reached by 3600 time steps and were maintained for the remainder of the full run to 4200 time steps. The initial condition and channel entrance profile was a straight line profile through the points 2y/D=0.0, U=81 m/s and 2y/D=1.0, U=155 m/s, as mentioned earlier.

A plot of the time averaged pressure versus along stream distance is shown in figure 4.5 with the analytic pressure profile from equation (3.6A) also plotted. The code results were time averaged over the last 100 time steps and spatially averaged across the width of the channel. One would not expect the pressure profile to be correct at the entrance to the channel since fully developed conditions do not exist there.

A typical snapshot of the instantaneous flow in the channel is shown in figure 4.6, where the fluid velocities at all grid points in the section x=0.4L to x=0.7L are shown.

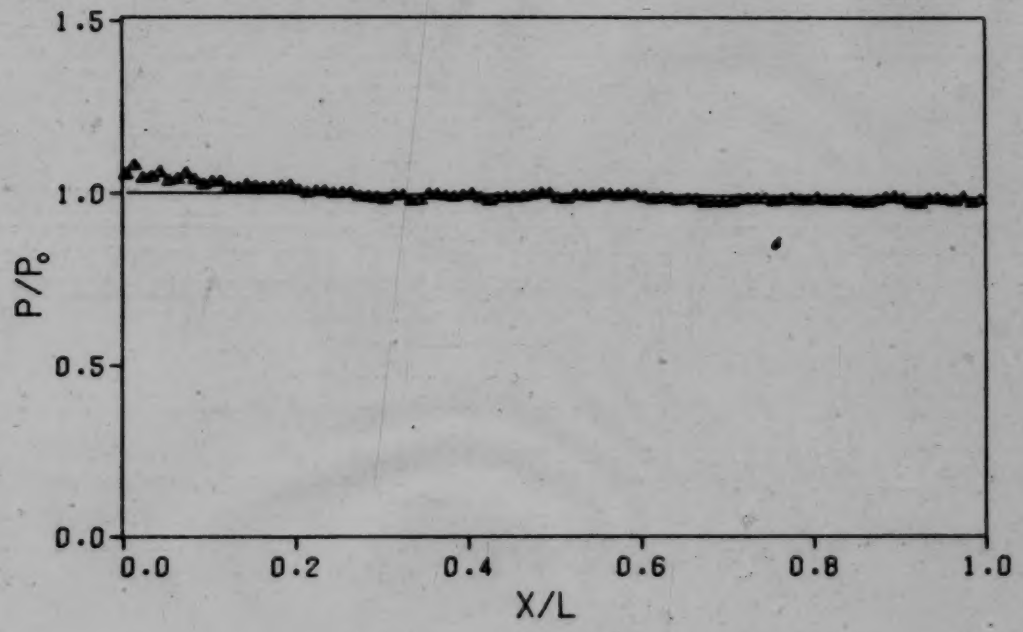

Figure 4.5 Comparison of pressure profile (triangles) to analytic results (straight line).

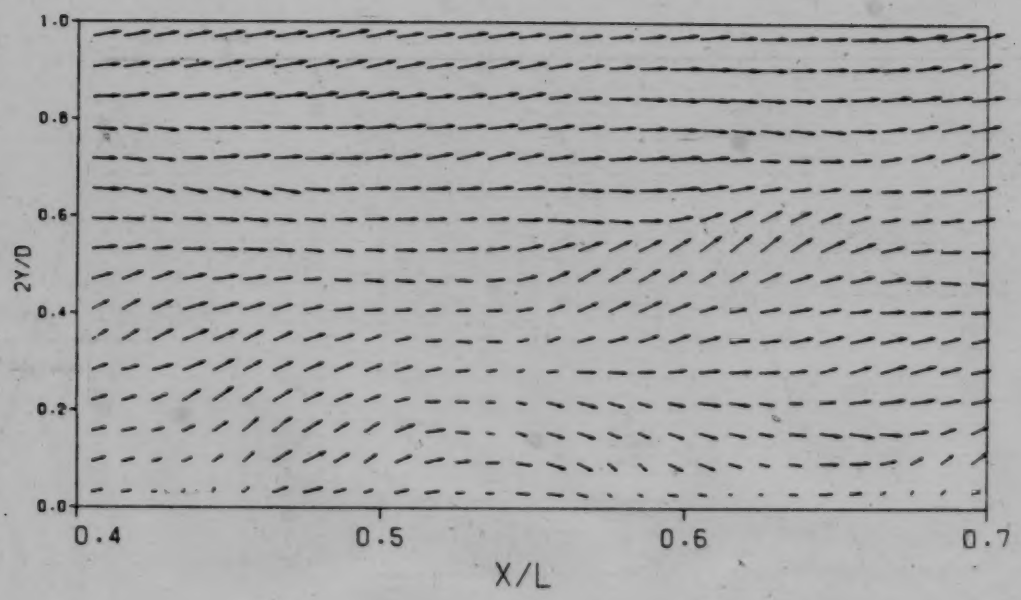

Figure 4.6 A typical instantaneous flow pattern. This snapshot is for 3100 time steps (3.3 ms). The arrows are scaled to the maximum velocity, which is 186.6 m/s (Note: <U>,=112.2 m/s)

In the simulation of the channel, the maximum number of 25 particles per cell was limited to 25 because of excessive computing costs required if more particles were used. With these values of n_o, the random particle noise mentioned earlier was of the same order as expected levels of turbulent fluctuations, so that turbulent correlations like $\langle u^2 \rangle$ and $\langle uv \rangle$ did not agree as well with experimental and analytic results. Code results for the shear stress, τ, as a function of cross stream distance were usually scattered about the straight line profile expected (equation [3.7]). The averaging used to obtain τ was the same as that used in obtaining U. Poor results are obtained because of random particle noise appearing in $\langle uv \rangle$. Values of $\langle u^2 \rangle^{1/2}/\langle U \rangle_s \cong 0.16$ and $\langle v^2 \rangle^{1/2}/\langle U \rangle_s \cong 0.23$ were obtained at 2400 time steps and values of $\langle u^2 \rangle^{1/2}/\langle U_s \rangle \cong 0.17$ and $\langle v^2 \rangle^{1/2}/\langle U_s \rangle \cong 0.25$ occurred at 3600 time steps, with 25 particles per cell. With 9 particles per cell, both turbulent intensities increased in time from $\langle u^2 \rangle^{1/2}/\langle U_s \rangle = 0.18$, $\langle v^2 \rangle^{1/2}/\langle U_s \rangle = 0.17$ at 600 time steps to 0.25, 0.30 at 1800 time steps. Values of $\langle u^2 \rangle^{1/2}/\langle U_s \rangle = 0.09$ and $\langle v^2 \rangle^{1/2}/\langle U_s \rangle = 0.07$ were expected from the discussion which yielded equation (3.9A). The code values for $\langle v^2 \rangle^{1/2}/\langle U_s \rangle$ are higher than $\langle u^2 \rangle^{1/2}/\langle U_s \rangle$ because the Gaussian averaging of fluid quantities was only done in the x-direction. As a result, noise levels were higher in the y-direction and also increased more rapidly in time.

However, after it was felt that the channel had reached a time step which was reasonably steady state with 25 particles per cell, a value of a_m=0.1 was used in the y-momentum equation (equation [A.2]) for the next 600 time steps. This was done to remove the large random noise levels appearing in the fluid velocity because of the random density fluctuations, mentioned in section 4.5. A value of a_m=0.0015 was still used in the x-momentum equation (equation [A.1]) and a value of a_t=0.5 was still used in the temperature equation. Within a few hundred time steps, random noise levels became very small and the shear stress profile, mostly determined by $-\rho\langle uv\rangle$, agreed well with the known profile, as is seen in figure 4.7. (Note: The computational cells next to the wall extended slightly beyond the viscous sublayer so that the viscous shear stress $\mu\partial U/\partial y$ was much smaller than the Reynolds shear stress $-\rho\langle uv\rangle$ for all cells except those next to the wall, as expected [section 3.2].) Values of $\langle u^2\rangle^{1/2}/\langle U_t\rangle$=0.11 and $\langle v^2\rangle^{1/2}/\langle U_t\rangle$=0.07 were observed at this point, which compares favorably with the expected values of 0.09 and 0.07. In the next 600 time steps, however, using such a large value of a_m=0.1 in the y-momentum equation interfered with the viscous terms, as mentioned earlier, and the code progressed to an incorrect solution.

Though there was too much noise in $\langle uv\rangle$ to obtain a proper shear stress profile with values of a_m=0.0015 in both the x and y-momentum equations and a_t=0.5; it is clear that

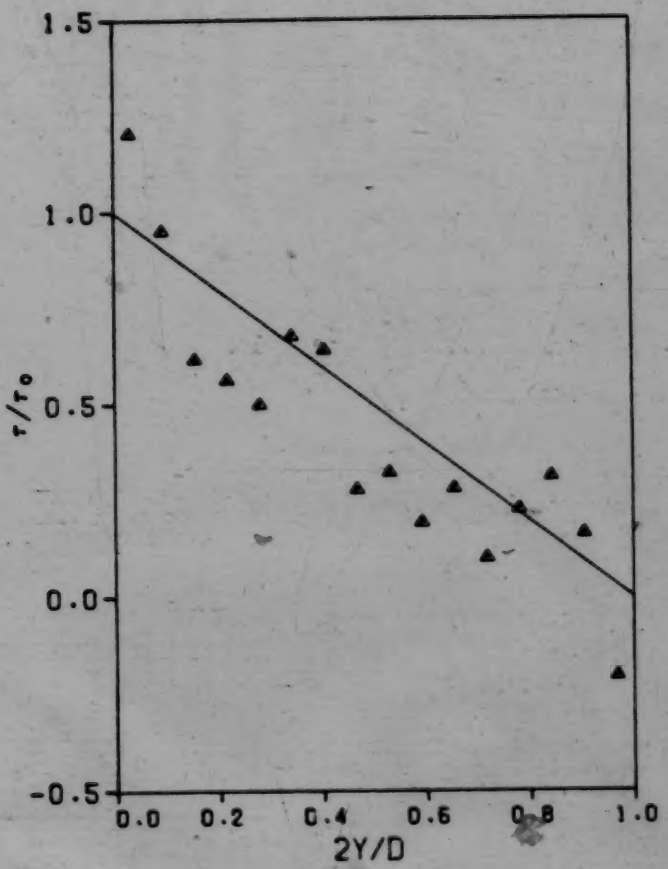

Figure 4.7 Comparison of code shear stress profile (triangles) to known profile (straight line).

the actual shear caused by $\tau = \mu \partial U/\partial y^{+} - \rho\langle uv \rangle$ was correct, since a correct velocity profile is observed and also a correct shear stress profile is seen if the noise is removed.

CHAPTER V

Simulation of a Magnetically Stabilized Gas Discharge

5.1 Analytic considerations

The initial purpose of this project was to develop a viscous fluid particle code for simulating a recently developed method of stabilizing a gas discharge, which was being experimentally tested at the University of Alberta. Several discharge geometries have been developed at the U of A, however, one which had been previously modelled by D. Antoniuk with an Alternating Direction Implicit (ADI) fluid code, as described in [4] and [16], was chosen so that the fluid particle approach could be compared to an Eulerian fluid approach in its ability to model a gas discharge.

The system that was modelled is described by D. Antoniuk in [16], and in more detail in [4]. These descriptions are summarized here for completeness. The discharge was a transverse electrode structure which was comprised of two copper plate electrodes with a d.c. magnetic field introduced by an electromagnet placed at the cathode, as depicted in figure 5.1. The magnetic field is shown in figures 5.2 and 5.3, which are also from [4] (note $B_\theta=0$). The maximum field strength in figure 5.3 was about 0.2 Tesla for the simulations done here. The gas in the discharge was a mixture of helium, nitrogen, and carbon dioxide. As described in [4] and [16], the total magnetic field ($\underline{B}$) of the discharge can be approximated very well as

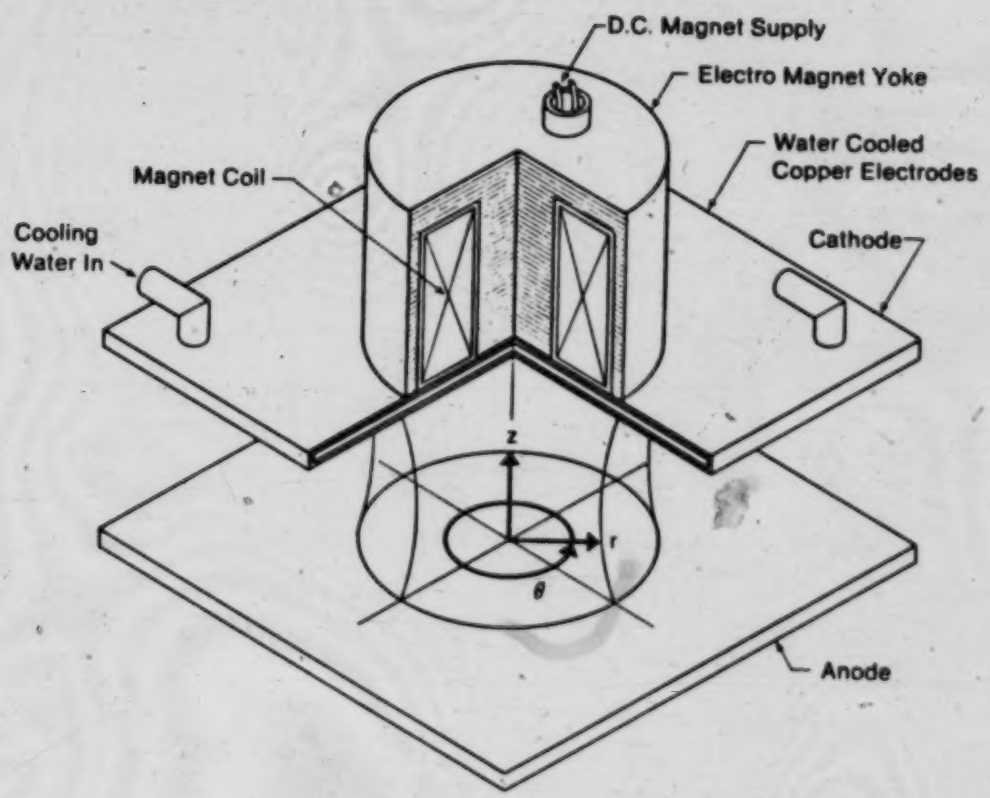

Figure 5.1 Magnetically stabilized transverse discharge geometry. This figure is taken from [4] with the author's permission.

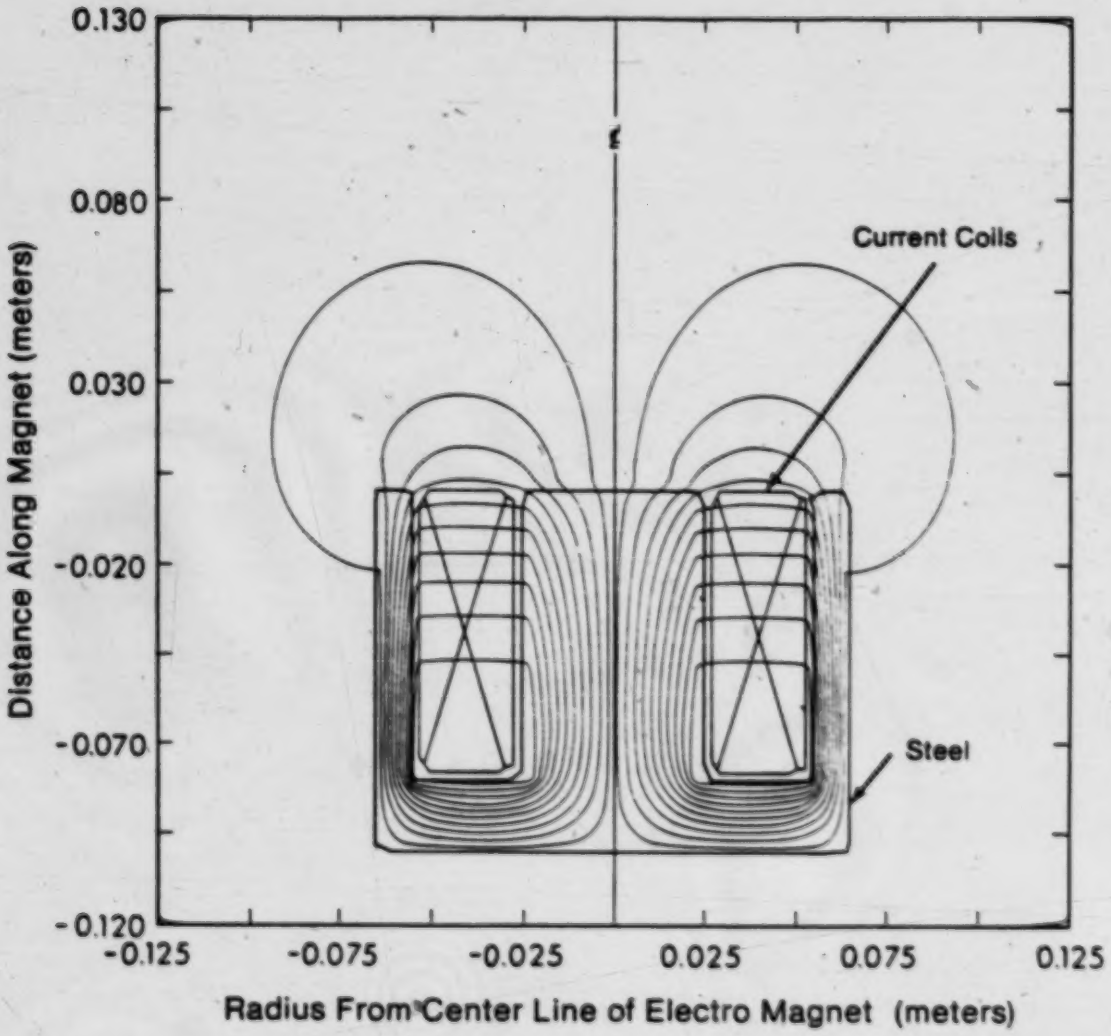

Figure 5.2 Magnetic flux density of electromagnet - figure taken from [4] with author's permission.

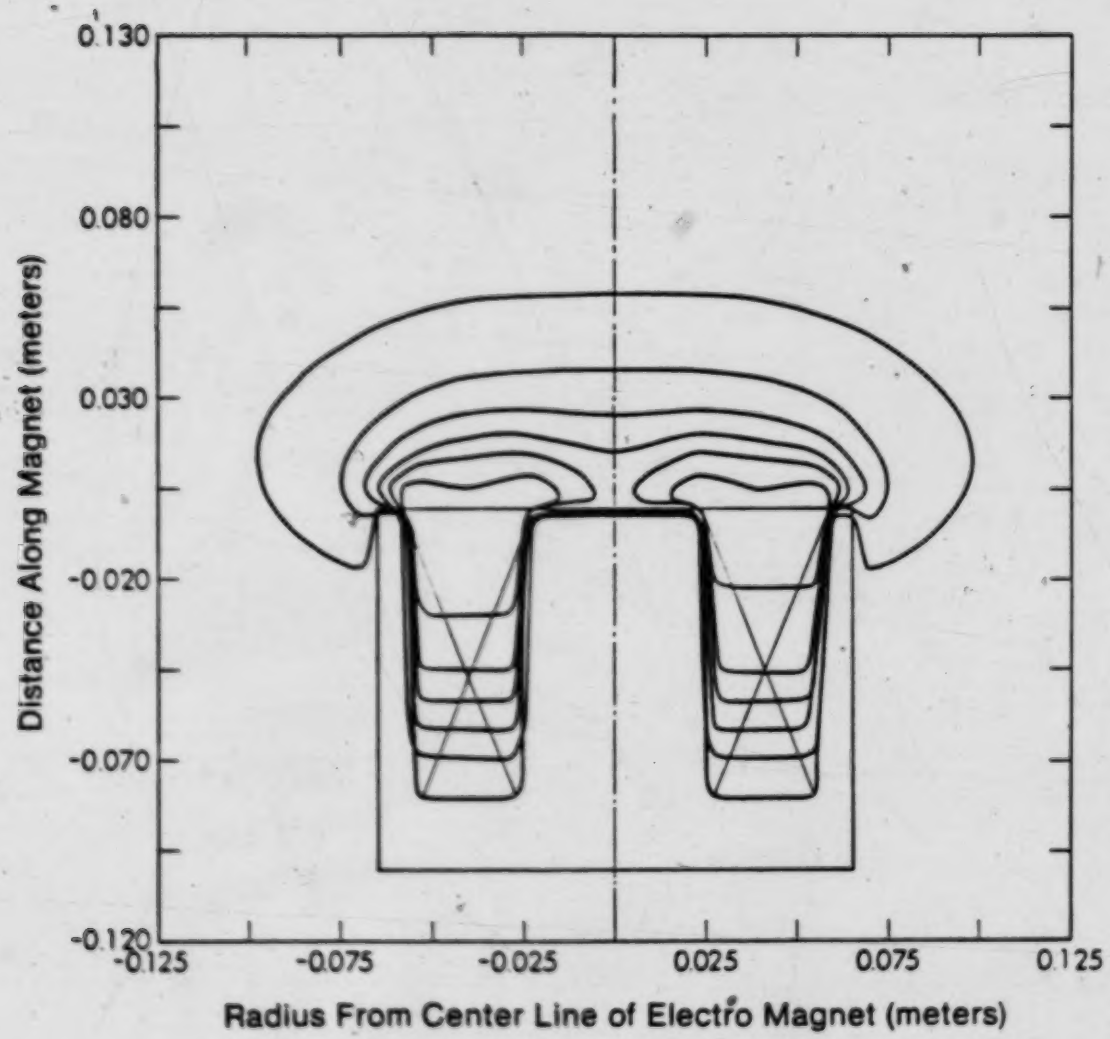

Figure 5.3 Magnetic field intensity contours of electromagnet - figure taken from [4] with author's permission.

being solely due to the electromagnet since virtually no magnetic induction is produced by the current flowing through the discharge. The current density flowing between the copper plates can be approximated by assuming it is axial, i.e. $J_\theta=J_r=0$, and as varying only in the radial direction. As in [4], a current density which was constant in time, of the form:

$$J_z(r) = J_0\exp[-(r-r_0)^2/2b^2]/b(2\pi)^{1/2} \qquad (5.1)$$

was used; where r_0 is the radial distance to where the peak current density occurs; $b^2=c^2/8\ell n2$ where c is the distance between the points where J_z is one-half its maximum value; and J_0 is such that the total current (I_t) flowing though the discharge ($I_t = \int\int J_z(r)rdrd\theta$) matches the current flowing in the wires to the electrodes. This profile is shown in figure 5.4. This corresponds to an annular glow discharge, which was observed experimentally by Antoniuk. In addition, a constant value of electric field, $\underline{E}$, with only an axial (z) component was used.

With these assumptions made, the complete MHD equations simplify considerably, since $\underline{J}$, $\underline{B}$ and $\underline{E}$ are no longer variables. Just as in channel flow, the fluid quantities $\underline{v}$, p, T, and ρ are the unknowns, and with only slight modifications to equations (2.2) and (2.6), the equations necessary for simulating this discharge can be written. It is necessary to add the $\underline{J}\times\underline{B}$ Lorentz force to the momentum equation (2.2) and the Joule heating term $\underline{J}\cdot\underline{E}$ to the temperature equation (2.6) [17]. One more simplifying

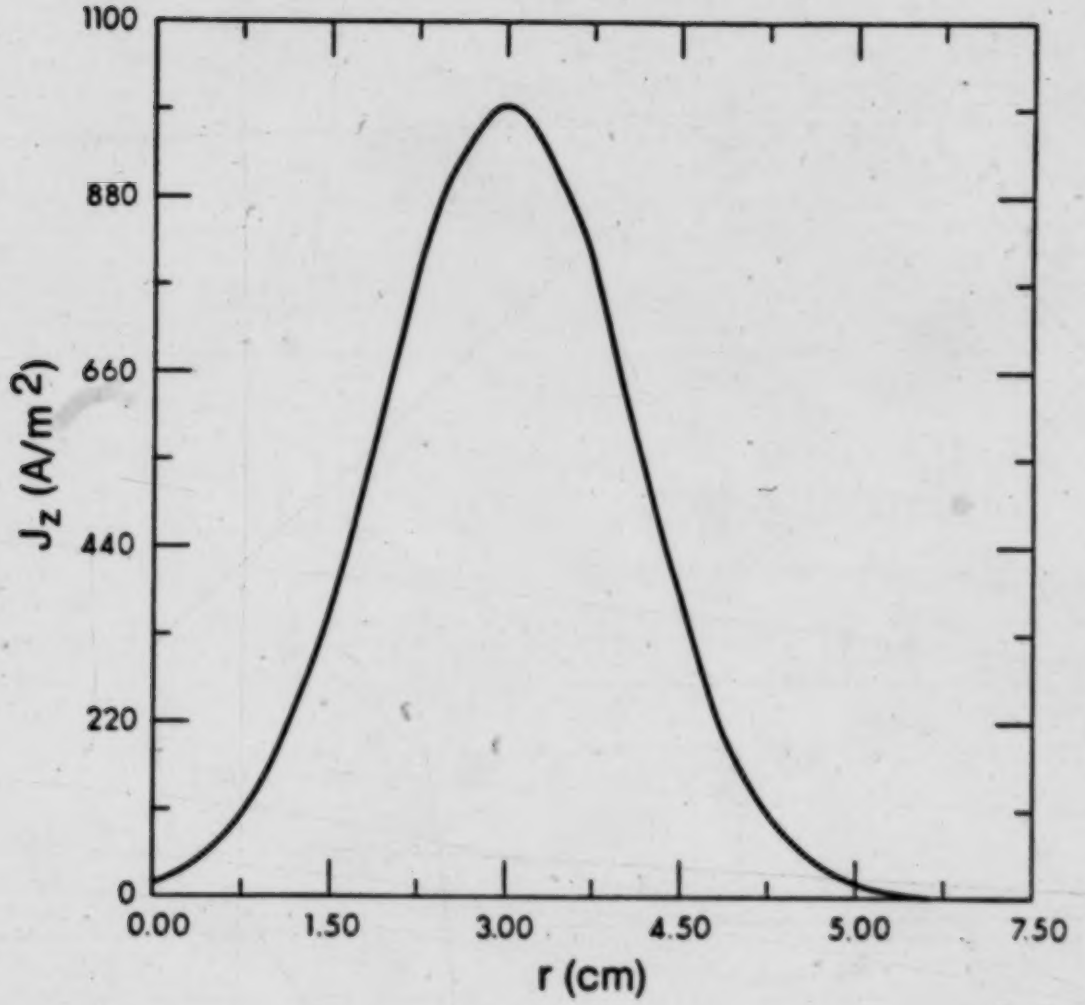

Figure 5.4 Gaussian shaped current density distribution - figure taken from [4] with author's permission

assumption can be made by noting that the Joule heating term will dominate the viscous dissipation ϕ from equation (2.6). This is because the ratio of viscous dissipation to thermal heating, defined as the Eckert number:

$$E \equiv V^2/(Cp)(\Delta T) \qquad (5.2)$$

(where V is the average magnitude of the total fluid velocity; Cp is the specific heat at constant pressure; and ΔT is the characteristic temperature difference), is much less than one for the gas being modelled here. The resulting equations for the magnetically stabilized gas discharge are written in cylindrical coordinates r and z in Appendix B. Because of the azimuthal symmetry involved, θ itself was not a variable, however, the azimuthal velocity v_θ was a variable i.e. $v_\theta = v_\theta(r,z)$, resulting in a so called "2&1/2 dimensional" model.

5.2 Computational considerations

5.2.1 Boundary and initial conditions

The computational aspect of this system is much simpler than the channel since there are no open flow boundaries. The computational window was chosen as illustrated in figure 5.5. Particles were reflected from the anode and cathode walls. In addition, at r=0 the particles were reflected since this is a line of symmetry. The radius, R, of the outer boundary of the window was chosen large enough, based on the results of Antoniuk, so that for r>R the state of the

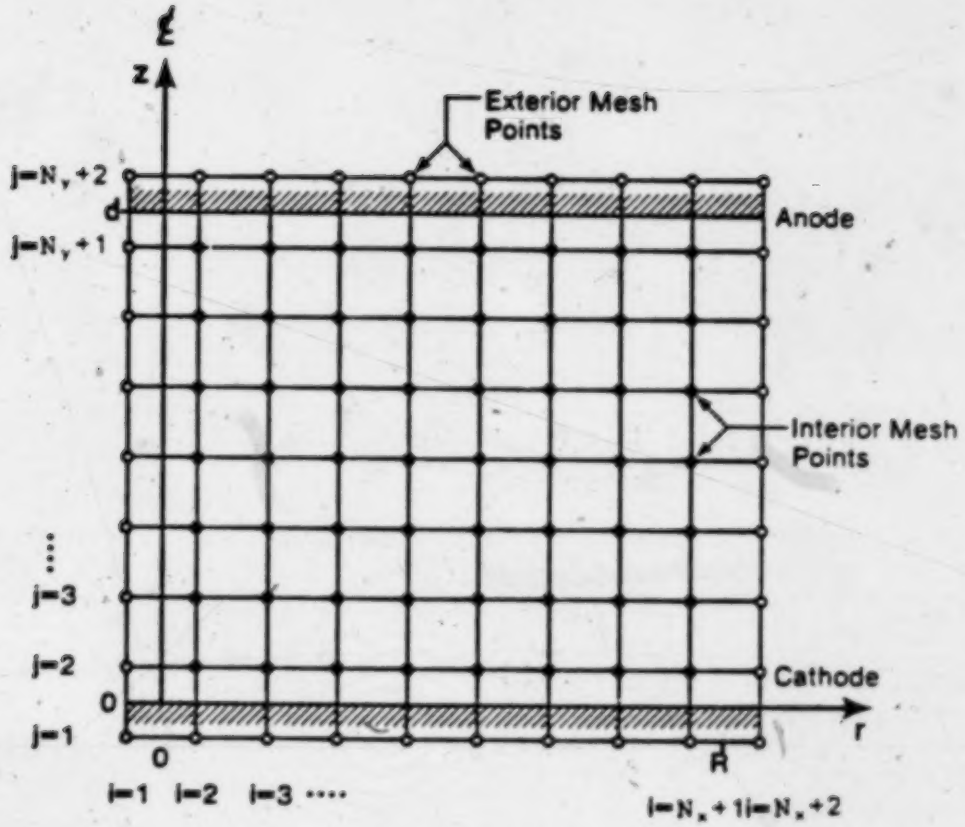

Figure 5.5 Arrangement of ghost point boundaries on the finite difference mesh - figure taken from [4] with author's permission

gas did not change significantly from its initial condition. Thus, near the outer boundary only very low fluid velocities were expected so that the particles were reflected from the r=R boundary; this kept the system from artificially losing too many particles, which had been a problem with the downstream boundary of the channel before pressure clipping was applied, as mentioned in Chapter 4. The fluid boundary conditions were chosen to be the same as those used in the ADI simulation of [4] and [16], in order that the results could be readily compared. On the r=0 boundary, a $\underline{v}$=0 condition was applied in [4] and [16], although it might seem more appropriate to use a Neumann condition for the axial velocity v_z and a homogeneous Dirichlet condition for v_θ and v_r. However, to be consistent with [4] and [16], $\underline{v}$=0 was used so that for the ghost points on the line r=-Δ/2: $\underline{v}_{i,j}=-\underline{v}_{i+1,j}$. Temperature and pressure were given Neumann conditions at this boundary. The ghost points on z=-Δ/2 satisfy $\underline{v}_{i,j}=-\underline{v}_{i,j+1}$, $T_{i,j}=2T_0 - T_{i,j+1}$, $p_{i,j}=p_{i,j+1}$, where j=1, so that on z=0: $\underline{v}$=0, T=T_0, $\partial p/\partial z$=0. Similarly the ghost points on z=d+Δ/2 were given fluid values which yielded: $\underline{v}$=0, T=T_0, $\partial p/\partial z$=0 for z=d. At r=R a Neumann condition was applied for pressure and temperature. Again, to be consistent with [4], the condition for velocity at r=R was: v_r=0, v_z=0 and $\partial v_\theta/\partial r$=0.

Initially, the gas was completely at rest with nine particles per cell, and all particles had a temperature of T_0 = room temperature. For simplicity, the phenomenological

parameters μ, C_v, κ, Cv of the gas were assumed constant, although it would be possible to use some known functional form (e.g. $\mu \propto T^{1/2}$) or a table lookup for each of these, however the fluid equations (2.2) and (2.6) would have to be rederived allowing for the temporal and spatial variation of these parameters. The simulated gas was a 2/8/10 Torr mixture of $CO_2/N_2/He$. The initial pressure p_o was thus 20 Torr, so that the initial density ρ_o was $\rho_o = p_o/RT_o$. (The R used here is the ideal gas constant, not to be confused with the R indicating the outer radius of the discharge.) The individual specific heats of N_2 and CO_2 were assumed constant at values obtained for a temperature of 500°C from [19]. Cv for He was taken as 3R/2. The specific heat of the mixture was then obtained by a mass weighted (gravimetric) average of the individual specific heats. The thermal conductivity of the mixture was estimated as $\kappa = 760 \times 10^{-4}$ J/sm°C for T=500°C based on tables of CO_2-N_2 mixtures and helium-other gas mixtures from [19]. (Note: using a gravimetric weighting for the thermal conductivity of mixtures is incorrect). The viscosity was chosen as $\mu = 6.6 \times 10^{-4}$ kg/ms as in [4] and [16]. A 19X11 computational grid window was used with grid spacing $\Delta = 5.1 \times 10^{-3}$m so that $R = 19\Delta$, $d = 11\Delta$. A total current (I_t) of five amps was used, with exactly the same current density distribution as shown in figure 5.4 i.e. $r_o = 3.0 \times 10^{-3}$ m, $c = 2.5 \times 10^{-3}$ m. A magnetic field which corresponded to a coil with the following characteristics was used: inner radius of windings = 2.5 cm; outer radius of

windings = 5.5 cm; length of coil = 7.5 cm; number of turns = 5600; current flowing through the coil = 2.0 A. The magnetic field values were obtained from Antoniuk, who used the computer program POTENT [20] to solve numerically for the magnetic field as a function of r and z. An electric field of E_z=-7130 V/m was used based on an experimental voltage drop of 400 V across the discharge [4]. The total voltage drop between the electrodes was about 525 V with I_z=5.0 A, but about 125 V occurred across the very thin nonlinear cathode fall region right next to the cathode. This region was ignored for the purpose of this simulation, since it does not obey the MHD equations, and was much smaller than a grid spacing. Δt was chosen as $\Delta t = 0.17\Delta/C_{s0}$ = 2.0 microseconds, where $C_{s0}=(\gamma RT_0)^{1/2}$. The particles were given a value of $a^2=0.5\Delta^2$, and the Gaussian fluid averaging used this same value of a^2. It should also be mentioned that a two dimensional Gaussian averaging was done for the fluid velocity and temperature i.e. the sums in equation (4.1) were extended to be over all grid points, not just over the cells in the jth row. Note that with $a^2=0.5\Delta^2$, the averaging for cell i,j is essentially only over cells i,j±1 and i±1,j, since f from equation (2.8) is 0.37 for these cells; while for i±1,j±1: f=0.14, and for i±2,j or i,j±2: f=0.02. A lower value of a^2/Δ^2 was used since it was found that noise levels were low enough with this value. (Note: without the Gaussian averaging, random noise levels about doubled)

5.2.2 Finite differencing considerations

All terms in the MHD equivalent to equations (2.2) and (2.6) were differenced either flux or force conservatively except the $\underline{v}\cdot(\nabla\cdot\underline{\underline{\tau}})$ term of the temperature equation. As with the channel, the additional computational effort involved in evaluating flux and force differencing for this term was not deemed worthwhile, since its effect (viscous work) was not of primary interest. Note that in the cells adjacent to the r=0 boundary, force conservative differencing cannot be used on terms of the form $(\partial F/\partial r)/r$ since a force conservative differencing would yield:

$$[(\partial F/\partial r)/r]_{i,j} = [(F_{i,j}-F_{i-1,j})/r_{i-1/2,j} + (F_{i,j+1}-F_{i,j})/r_{i-1/2,j}]/2\Delta$$

but $r_{i-1/2,j}=0$ for these cells, so that $1/r_{i-1/2,j}$ is undefined. For such terms where F was a single variable, $(F_{i+1,j}-F_{i-1,j})/r_{i,j}\Delta$ was used. If F=rG, where G=G($\underline{v}$,p,T), then the term was differentiated i.e. $(\partial F/\partial r)/r = G/r + \partial G/\partial r$ where $\partial G/\partial r$ was differenced flux conservatively, and $G/r = G_{i,j}/r_{i,j}$.

5.3 Results and discussion

A run of 2000 time steps (4 ms) was done using the boundary and initial conditions mentioned earlier. A contour plot of v_θ is shown in figure 5.6. Previous time steps have very similar contour plots for v_θ except that the value of v_θ increases in time, so that they are essentially scaled versions of that shown in figure 5.6.

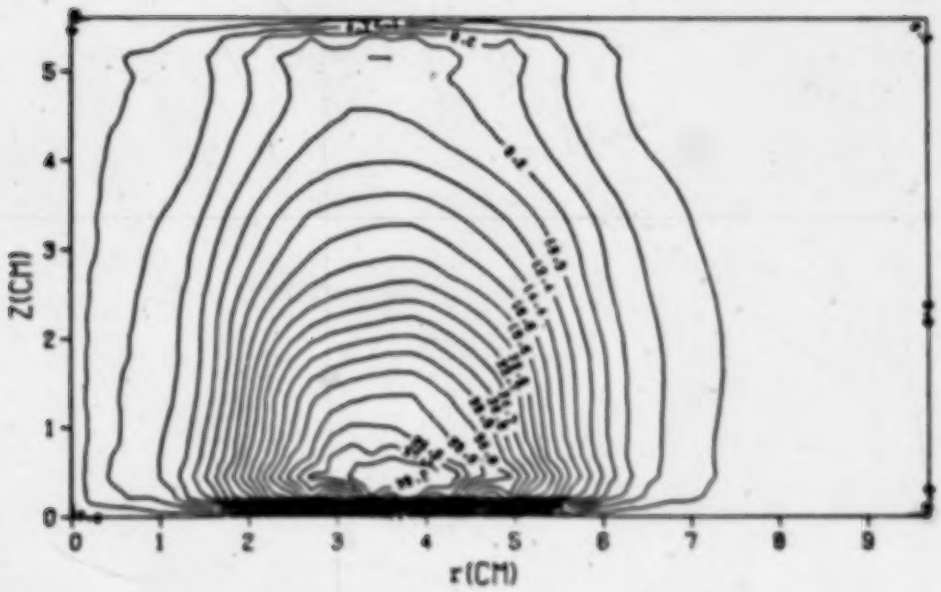

Figure 5.6 Contour plot of v_θ in m/s from particle code at t=4 ms (2000 time steps).

Unfortunately, because we have assumed θ symmetry, any Rayleigh-Taylor instabilities [17] in θ which might arise in the actual discharge cannot occur in the simulation. What is meant by a Rayleigh-Taylor instability is shown in figure 5.7. The initial position of a section of plasma is indicated by the circle. The circumstances under which this instability will occur are those where a heavier fluid is "supported" by a lighter fluid. In this case, a hot, lower density region is assumed to be in equilibrium with a denser, lower temperature region, while a centrifugal force pushes the heavier fluid onto the lighter one. The instability causes portions of the discharge to flow radially outward (indicated by the arrows pointing radially outward) while neighboring sections flow radially inward. This tends to mix the plasma so that any variations in the state of the plasma are diffused. The wavy line indicates the position of the same plasma mass shortly after the onset of the instability. Note also that no θ mixing or θ turbulence is modelled in a 2&1/2 dimensional model, so that the computational system is artificially too "calm".

In our simulation, the Joule heating term $J_z E_z$ produces a temperature profile which is reasonably similar in shape to that of J_z (since E_z is a constant). Such a temperature profile coupled with the centrifugal force resulting from v_θ would certainly cause Rayleigh-Talor instabilities to form; but, with just the two spatial dimensions r and z, it is impossible to simulate these instabilities, so that the

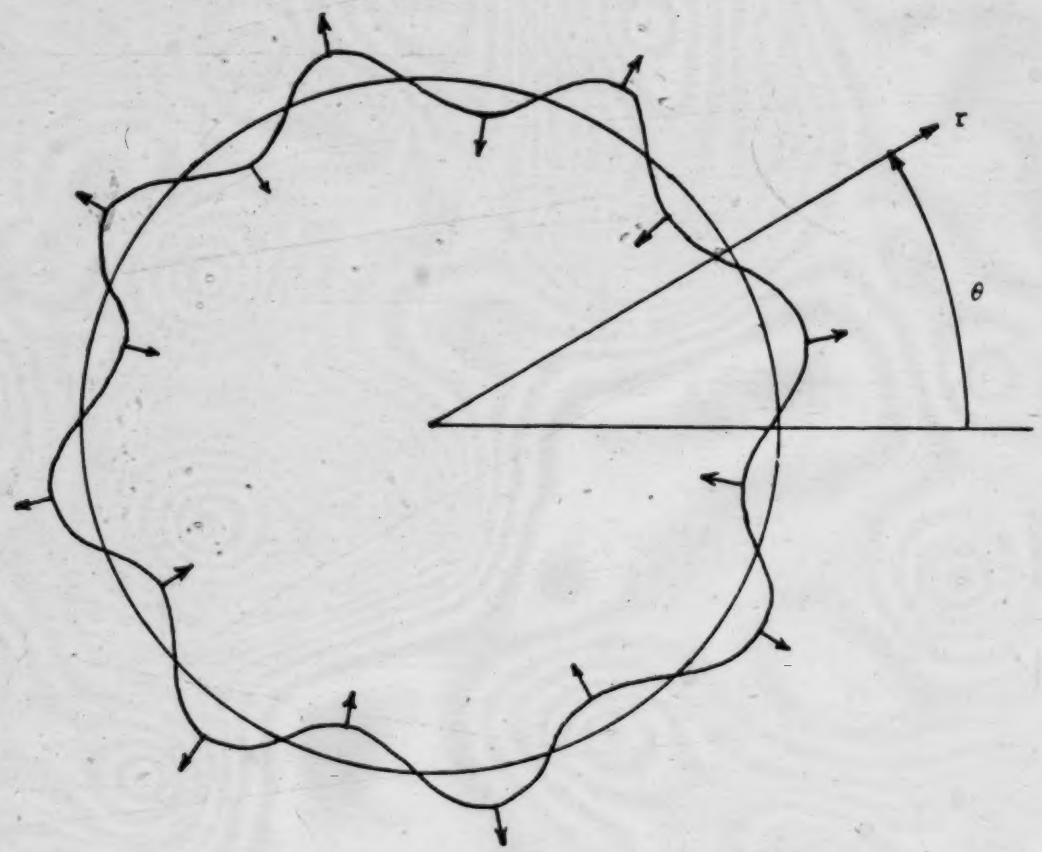

Figure 5.7 A Rayleigh-Taylor instability, as shown, would normally result in the gas discharge geometry where a less dense fluid contains a more dense fluid. See p. 95 for an explanation.

Joule heating term would heat a ring of plasma to very high temperatures, while the r=R wall is held at room temperature. Clearly, this is not a physical situation. Thus, to obtain a useful result from our 2&1/2 dimensional simulation, it is necessary to stop the code after a number of time steps and use the results of this time step as an approximation to the steady state results.

It was found that if a simulation was stopped after t_{mx}=4 ms, where t_{mx} is the time required for the average gas temperature to increase from its initial value of room temperature to a value in the range of experimental values from [4], then v_θ (which at t=0 was zero) was also in the range of experimental data. The critical variable for which good experimental data exists [4] is the peak value of v_θ ($v_{\theta mx}$). It was also found that if temperature was not used as a variable, then the amount of time required for $v_{\theta mx}$ to reach a steady state value was also about 4 ms.

The particle code results correspond very well with those obtained by D. Antoniuk both experimentally and computationally in [4]. Antoniuk's code was an isothermal ADI (Alternating Direction Implicit) Eulerian fluid code which assumed a constant temperature of 425 K. His computational results indicate a steady state value of $v_{\theta mx}$=35 m/s occurring after 4 ms. A contour plot of the steady state (t=4 ms) sheared rotational velocity distribution produced by his ADI program is shown in figure 5.8. Experimentally, Antoniuk observed a steady state value of

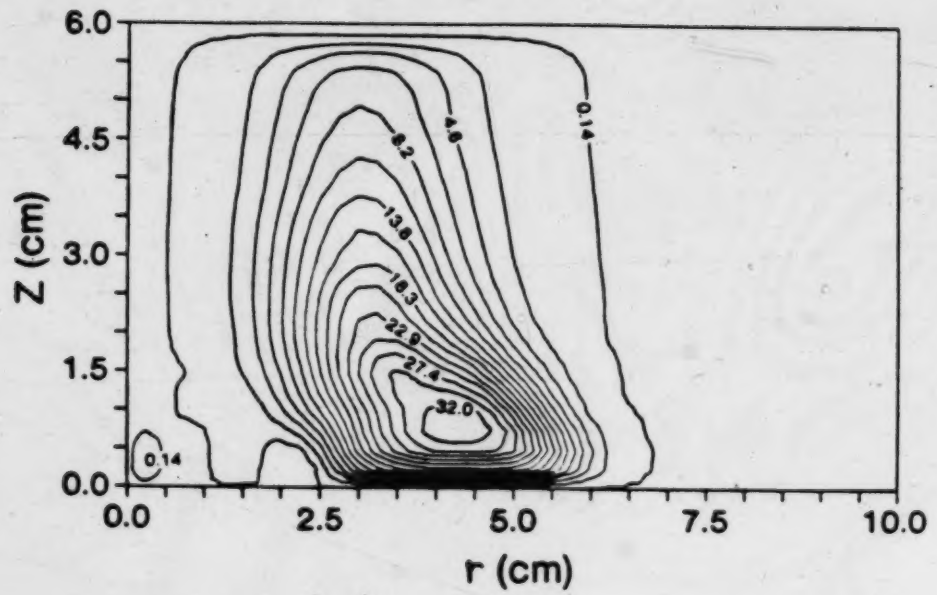

Figure 5.8 Contour plot of v_θ at t=4 ms obtained by Antoniuk [16] using ADI fluid code.

$v_{\theta mx}$=38.5 m/s [4, p. 168] with a total discharge current of I_t=5.0 A and a magnetic field associated with the magnetic coil current I_m=2.0 A. (Note that in [4], two equivalent coils were used; one required 2.0 A, as mentioned here, to give the same field used in the particle code, while the other required 4.0 A to give the same field. The data on [4, p. 168] for $v_{\theta mx}$ uses the 4.0 A coil. All I_m values listed in the work herein are for the 2.0 A coil.)

Antoniuk recorded maximum values of v_r in the range of 10 to 20 m/s [4, p. 174-177] for a 1/7/12 CO_2/N_2/He mixture. A typical velocity flow plot of fluid velocity $\underline{\hat{v}}_{i,j}$ (from equation [4.1]) is shown in figure 5.9, where the maximum velocity is 36.4 m/s. Antoniuk also found that a circular eddy developed in his simulation. This eddy covered the entire computational window in the r-z plane of his isothermal ADI simulation. Although secondary flow developed with the fluid particle simulation, no such large scale vortex developed. Isothermal runs were done to determine if the temperature profile stopped the vortex from forming, but no such large scale vortex developed in either isothermal or nonisothermal runs. It is difficult to say whether such a vortex was observed experimentally. Since Antoniuk's ADI simulation had difficulty dealing with reversals in flow velocities occurring between cells, it is possible that the vortex structure was a computationally preferred flow mode of the ADI simulation.

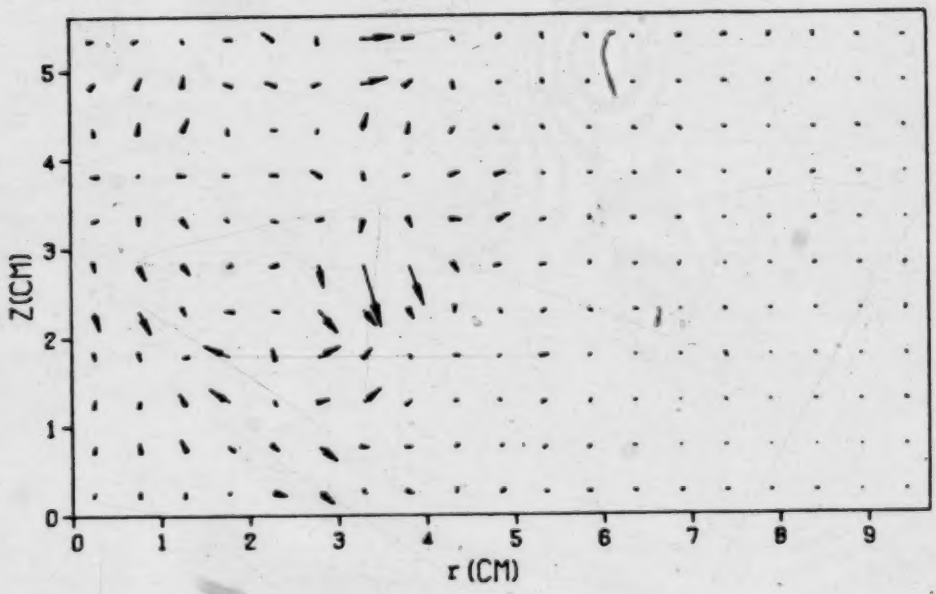

Figure 5.9 Typical particle code velocity flow plot. The arrows length are scaled with the magnitude of velocity. The maximum velocitiy is 36.4 m/s.

It is difficult to compare any other code results to Antoniuk's results because most of Antoniuk's data was obtained for considerably different discharge conditions. He obtains a temperature profile [4, p. 217] for a discharge current of I_t=4.0 A, but using a cathode which was composed of many metal pins, not a flat plate. The multipin cathode causes the temperature profile to be less peaked than for the flat plate cathode. In addition, a discharge current of I_t=4.0 A produces a discharge with a current density that has its peak value in a different place than that in figure 5.4. As well, there will be less heating with only 4.0 A flowing through the discharge. Even so, one can estimate that the temperatures obtained from the code at t=4 ms, as in figure 5.10, are quite reasonable, as mentioned earlier.

It should be noted that the value of multistreaming coefficient a_m used for this geometry was very much larger than that used in the channel. The value of a_m=0.1 used here can produce artificial drag forces which are up to several thousand times larger than the viscous forces caused by μ. (An analysis similar to that which yielded equation [4.9] yields a value of $a_m \cong 0.0001$ here.) This is in contrast to the channel where it was necessary to set a_m so that artificial drag forces and viscous forces were of the same order. As mentioned in chapter 4, it is not necessary to restrict a_m in this way since there is no boundary layer flow in the r-z plane and the flow in the θ direction is artificially laminar because the code is two dimensional.

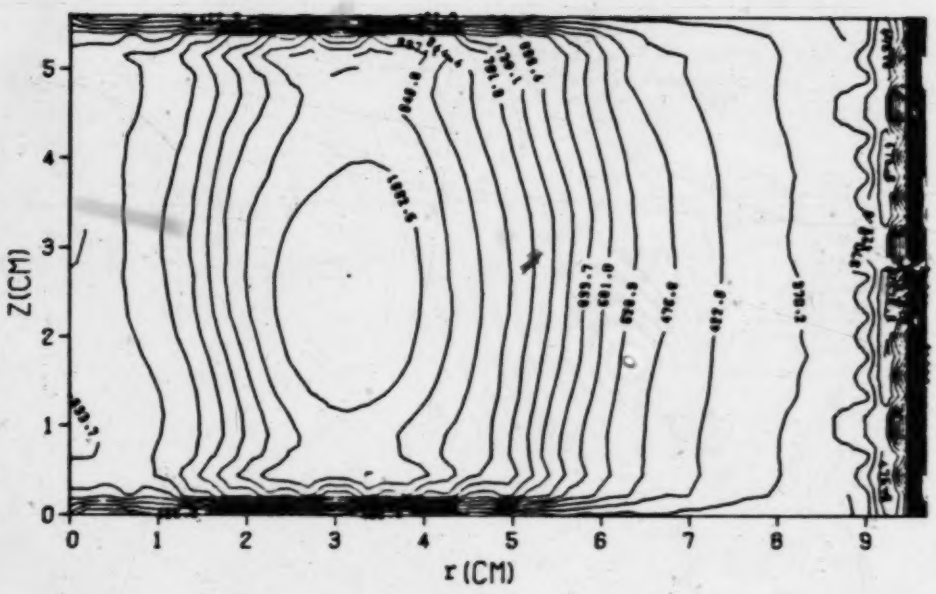

Figure 5.10 Contour plot of gas temperature T in degrees Kelvin from particle code at t=4 ms.

Thus, there is no need to have intracell and intercell viscous effects compatible so that there is no difficulty in using a value similar to the value of $\alpha_t=\alpha_m=0.5$, used by Leboeuf et al. in [2] for an inviscid fluid.

It should also be mentioned that if a conventional ∇p finite differencing was used, values of the turbulent kinetic energy $\langle q^2\rangle^{1/2}$ (as in equation [3.10]) increased by a factor of ten or more from the modified ∇p case to levels that produced very unrealistic results. This increase in $\langle q^2\rangle^{1/2}$ was due to large scale random motions, as mentioned in section 2.2.2.1, and was only kept stable by the Gaussian averaging combined with the multistreaming term α. Using the modified ∇p differencing and values of $\alpha_m=0.1$, $\alpha_t=0.5$, random particle noise levels were small.

CHAPTER VI

Concluding Remarks

6.1 Summary

An approach for applying a NGP (nearest grid point) finite particle method to viscous, compressible fluids has been described. The fluid is modelled very simply by a set of discrete finite particles. The particles carry a value of velocity and temperature, which give rise to fluid velocities and temperatures on an Eulerian grid by averaging the particle velocities and temperatures in each cell on the grid. The particles also have a Gaussian mass density shape, allowing the fluid density to be calculated by summing the particle densities using a NGP approach. The instantaneous fluid equations were written in a Lagrangian conservative form so that the particle temperatures and velocities were pushed conservatively in time using the Lagrangian derivatives of velocity and temperature, which were known by finite differencing the fluid equations. The ∇p term in the velocity equation was dealt with in a special manner which removed the previous difficulty of excessively high random particle noise levels appearing with the NGP approach. In the channel flow simulation, a small value for the multistreaming coefficient a_m (appearing in the momentum equation) was necessary otherwise an incorrect velocity profile developed. Analytic and experimental results for channel flow were illustrated. A particle model was

developed for turbulent channel flow, with the use of buffer zones at the open ends and pressure clipping at the downstream end to maintain a steady flow rate. The constraints on various parameters necessary for a convergent, stable solution were developed. To maintain stability, it was found that the temperature equation was required for channel flow. Excellent agreement with experimental results for the mean value of pressure and along stream velocity for turbulent channel flow (Reynolds number=5333) were obtained. Profiles of the turbulent correlation <uv> agreed well with analytic results if the random noise was reduced to low levels, otherwise <uv> was covered by noise which was implicit to this average. The noise however did not cause turbulent transport of momentum, as seen by the correct velocity profile and correct shear stress profile.

A magnetically stabilized gas discharge was also modelled. Good results were obtained in comparison with the experimental data of Antoniuk [4] and with the computational results of Antoniuk's isothermal ADI conventional Eulerian fluid code.

6.2 Some further remarks

The NGP particle approach to modelling a fluid is a very straightforward method that is easily understood and which is also very physical in how it deals with the fluid. As a result, difficulties with the method can usually be

understood on a very practical level without requiring much detailed analysis. Because each cell has many particles, the resolution of a flow is also greater than that of conventional fluid methods.

A typical run for the channel required about 75 (89) minutes of computing time per 600 time steps, for 9 (25) particles per cell, but 85% (70%) of this computing time was taken up by the subroutines used for calculating the Gaussian densities and Gaussian averaging. Because these were done on the array processor (FPS 190L), which cost 1/20 that of the mainframe computer (Amdahl 5860 which is equivalent to an IBM 370), the cost of modelling the gas discharge was only one and one half to two times that of Antoniuk's ADI simulation. With the gas discharge, a typical run required about seven minutes mainframe CPU time and 18 minutes of array processor time.

6.3 Future applications and considerations

Clearly, there is much room for future applications of the fluid particle method. The work herein has developed necessary considerations when dealing with turbulent viscous compressible fluids in both open and closed systems using a fluid particle approach. It would be most interesting to determine how closely more of the details of actual turbulence are modelled by a fluid particle code. It would of course be necessary to remove the random noise from the turbulence. If enough particles are used per cell in a NGP

approach, the random particle noise appearing for small a_m should be reduced to low levels, as indicated by equations [4.15] and [4.16]. Because the Gaussian subroutines take more than 70% of the computing time and because they depend on the number of cells, and not the number of particles per cell (n_o), the total computing time increases by only a few percent by doubling the number of particles per cell. Unfortunately, at the U of A, this increase was in mainframe CPU time and not AP time, so that costs increased by a factor of about 1.5 for n_o doubling. On a vector processing supercomputer (e.g. a Cyber 205 or Cray 1) one would likely put the whole program on the supercomputer so that runs with large n_o would require only a small percentage more time. It is felt that noise levels in the channel would likely be reduced to small levels if at least 81 particles per cell were used.

Other methods of grid particle assignment might be attempted to reduce noise levels, but these would likely require many special considerations unique to each method. The noise levels in the area weighting method listed in [7] would have to be reduced before being usable in turbulent flow, while the subtracted dipole approach (SUD) described in [7] would require special modification for nonperiodic flow because it uses fast Fourier transforms (FFT) for the spatial derivatives, so that leakage and possibly aliasing errors would appear.

Areas for further work include: the simulation of a viscous flow past an airfoil; extension of the fluid particle method to incompressible fluids; the development of a NGP approach with variable grid spacings; the use of coordinate transformation techniques to simulate irregular geometries or to transform a variably spaced grid into a regularly spaced one and application of a fluid particle approach to this set of equations.

In conclusion, as Hockney and Eastwood comment [21, p. 347], "the principle of moving the fluid as particles in order to ensure positive density and mass conservation, and calculating the force on the fluid particles from field quantities derived on a fixed mesh, is sound and likely to have wide application in fluid dynamics."

References

[1] H. Schlichting, "Boundary-Layer Theory", McGraw-Hill, New York, 1979, 7th edition.

[2] J.N. Leboeuf, T. Tajima and J.M. Dawson, "A Magnetohydrodynamic Particle Code for Fluid Simulation of Plasmas", J. Comput. Phys., vol. 31, pp. 379-408, 1979.

[3] C.H. Finan, Ph.D. Thesis, University of California, Davis, Department of Applied Science, 1980.

[4] D.M. Antoniuk, Ph.D. Thesis, University of Alberta, Edmonton, Department of Electrical Engineering, 1983.

[5] D. Potter, "Computational Physics", Wiley, New York, 1973.

[6] J.M. Dawson, "Particle simulation of plasmas", Reviews of Modern Physics, vol. 55, no. 2, April 1983.

[7] F. Brunel, J.N. Leboeuf, T. Tajima and J.M. Dawson, "Magnetohydrodynamic Particle Code: Lax-Wendroff Algorithm with Finer Grid Interpolations", J. Comput. Phys., vol. 43, pp. 268-288, 1981.

[8] A. Nishiguchi and T. Yabe, (Institute of Laser Engineering, Osaka University, Yamada-Oka, Suita, Osaka 565, Japan), "Second Order Fluid Particle Scheme", preprint of a paper to be published.

[9] F.M. White, "Viscous Fluid Flow", McGraw-Hill, New York, 1974.

[10] R.J. Mason, "Implicit Moment PIC-Hybrid Simulation of Collisional Plasmas", J. Comput. Phys., vol. 51, no. 3, pp. 484-501, Sept. 1983.

[11] J.O. Hinze, "Turbulence", McGraw-Hill, New York, 1975, 2nd edition.

[12] S. Goldstein, "The Similarity Theory of Turbulence, and Flow between Parallel Planes and through Pipes", Proc. Roy. Soc. of London A, vol. 159, pp. 473-506, 1937.

[13] R. Techo, R.R. Tickner and R.E. James, "An Accurate Equation for the Computation of the Friction Factor for Smooth Pipes from the Reynolds Number", J. Appl. Mechanics, vol. 87, pt. 3, p. 443, June 1965.

[14] D.J. Wilson, class notes from Mechanical Engineering 632 (An Introduction to Turbulence) taught by Professor Wilson at the University of Alberta in 1983.

[15] David L. Book (editor), "Finite-Difference Techniques for Vectorized Fluid Dynamics Calculations", Springer-Verlag, New York, 1981.

[16] D.M. Antoniuk, C.E. Capjack and H.J.J. Sequin, "Computer Simulation of Gas Transport in a Magnetically Stabilized Glow Discharge", J. Appl. Phys., vol. 55, no.3, pp. 708-713, Feb. 1984.

[17] Francis F. Chen, "Introduction to Plasma Physics", Plenum Press, New York, 1974.

[18] J.H. Keenan and J. Kaye, "Gas Tables", Wiley, New York, 1948.

[19] N.B. Vargaftik, "Tables on the thermophysical properties of liquids and gases", Hemisphere, Washington, 1975, 2nd edition.

[20] C.L. Thomas, "Software for Numerical Mathematics", Lougborough Leic. Eng., Academic Press, p. 315, 1974.

[21] R.W. Hockney and J.W. Eastwood, "Computer Simulation Using Particles", McGraw-Hill, London, 1982.

[22] William D. Stanley, "Digital Signal Processing", Reston Publishing Company, Reston U.S.A., 1975.

Appendix A

The full compressible fluid equations (2.2) and (2.5) in two dimensional cartesian coordinates x and y are:

$$\rho dv_x/dt = -\partial p/\partial x + \mu[\partial^2 v_x/\partial x^2 + \partial^2 v_x/\partial y^2] + \mu[\partial^2 v_x/\partial x^2 + \partial^2 v_y/\partial x\partial y]/3 \qquad (A.1)$$

$$\rho dv_y/dt = -\partial p/\partial y + \mu[\partial^2 v_y/\partial x^2 + \partial^2 v_y/\partial y^2] + \mu[\partial^2 v_x/\partial x\partial y + \partial^2 v_y/\partial y^2]/3 \qquad (A.2)$$

$$\rho Cv(dT/dt) = \kappa[\partial^2 T/\partial x^2 + \partial^2 T/\partial y^2] - \partial(pv_x)/\partial x - \partial(pv_y)/\partial y + \mu\{v_x(\partial^2 v_x/\partial x^2+\partial^2 v_x/\partial y^2) + v_y(\partial^2 v_y/\partial x^2+\partial^2 v_y/\partial y^2) + [v_x(\partial^2 v_x/\partial x^2+\partial^2 v_y/\partial x\partial y) + v_y(\partial^2 v_y/\partial y^2+\partial^2 v_x/\partial x\partial y)]/3 + 4[(\partial v_x/\partial x+\partial v_y/\partial y)^2 - (\partial v_y/\partial y)(\partial v_x/\partial x)]/3 + (\partial v_y/\partial x+\partial v_x/\partial y)^2\} - \rho v_x dv_x/dt - \rho v_y dv_y/dt \qquad (A.3)$$

The viscous term in the temperature equation (A.3) can be written in a flux conservative form (see Chapter 2), however the rewards of such a form were not believed to be worth the excess complexity added to the finite differencing since the viscous work and dissipation were not of major concern, as mentioned in Chapter 2.

Appendix B

The full compressible fluid equations (2.2) and (2.6) written for two cylindrical coordinates r and z and three velocity components v_r, v_θ, and v_z are:

$$\rho dv_r/dt = -\partial p/\partial r + \frac{\mu}{r}[\frac{\partial}{\partial r}(r\partial v_r/\partial r)] + \mu[\partial^2 v_r/\partial z^2 - v_r/r^2]$$
$$+ \mu\frac{\partial}{\partial r}[\frac{1}{r}\partial(rv_r)/\partial r]/3 + \mu(\partial^2 v_z/\partial z\partial r)/3$$
$$+ (J_\theta B_z - J_z B_\theta) + \rho v_\theta^2/r \qquad (B.1)$$

$$\rho dv_\theta/dt = \frac{\mu}{r}[\frac{\partial}{\partial r}(r\partial v_\theta/\partial r)] + \mu[\partial^2 v_\theta/\partial z^2 - v_\theta/r^2]$$
$$+ (J_z B_r - J_r B_z) - \rho v_\theta v_r/r \qquad (B.2)$$

$$\rho dv_z/dt = -\partial p/\partial z + \frac{\mu}{r}[\frac{\partial}{\partial r}(r\partial v_z/\partial r)] + (4\mu\partial^2 v_z/\partial z^2)/3$$
$$+ \frac{1\mu}{3r}\partial^2(rv_r)/\partial r\partial z + (J_r B_\theta - J_\theta B_r) \qquad (B.3)$$

$$\rho Cv(dT/dt) = -\rho v_r dv_r/dt - \rho v_\theta dv_\theta/dt - \rho v_z dv_z/dt$$
$$+ \frac{\kappa\partial(r\partial T/\partial r)}{r\partial r} + \kappa\partial^2 T/\partial z^2$$
$$- \{[\frac{\partial}{\partial r}(rpv_r)]/r + \partial(pv_z)/\partial z\}$$
$$+ \mu[v_r(\nabla^2\underline{v})_r + v_\theta(\nabla^2\underline{v})_\theta + v_z(\nabla^2\underline{v})_z$$
$$+ (v_r\partial^2 v_z/\partial z\partial r + v_z\partial^2 v_r/\partial z\partial r - v_r\partial^2 v_r/\partial z^2)/3$$
$$+ (v_z\partial v_r/\partial z)/r]$$
$$+ J_r E_r + J_\theta E_\theta + J_z E_z \qquad (B.4)$$

where $\nabla^2\underline{v} = (\nabla^2\underline{v})_r\hat{e}_r + (\nabla^2\underline{v})_\theta\hat{e}_\theta + (\nabla^2\underline{v})_z\hat{e}_z$ and the $\hat{e}$'s are unit vectors.